

NPS52-86-002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



Surface Construction from Planar Contours

Patrick G. Hogan

Michael J. Zyda

January 1986

Approved for public release; distribution unlimited

Prepared for:

Chief of Naval Research
Arlington, VA 22217

FedDocs
D 208.14/2
NPS-52-86-002

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral R. H. Shumaker
Superintendent

D. A. Schradly
Provost

The work reported herein was supported by in part by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by:

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY CA 93943-5101

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-86-002	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Surface Construction from Planar Contours		5. TYPE OF REPORT & PERIOD COVERED
7. AUTHOR(s) Patrick G. Hogan Michael J. Zyda		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		8. CONTRACT OR GRANT NUMBER(s) 61152N; RR000-01-NP N0001485WR41005
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE January 1986
		13. NUMBER OF PAGES 47
		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) surface construction surface triangulation planar contours		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Many scientific and technical endeavors require the reconstruction of a three-dimensional solid from a collection of two-dimensional contours. One method for this reconstruction involves a procedure whereby individual pairs of contours are mapped together to form triangular surface patches. In this paper, we present an algorithm which not only handles mapping situations of simple, closed contours but also mappings of multiple contours per plane and partial contour mappings. Also included is a discussion of the algorithm's limitations and heuristics.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Surface Construction from Planar Contours ‡

Patrick G. Hogan and Michael J. Zyda

Naval Postgraduate School,
Code 52, Department of Computer Science,
Monterey, California 93943-5100

ABSTRACT

Many scientific and technical endeavors require the reconstruction of a three-dimensional solid from a collection of two-dimensional contours. One method for this reconstruction involves a procedure whereby individual pairs of contours are mapped together to form triangular surface patches. In this paper, we present an algorithm which not only handles mapping situations of simple, closed contours but also mappings of multiple contours per plane and partial contour mappings. Also included is a discussion of the algorithm's limitations and heuristics.

Categories and Subject Descriptors: I.3.3 [Picture/Image Generation]: display algorithms; I.3.5 [Computational Geometry and Object Modeling]: surface and solid representations; I.3.7 [Three-Dimensional Graphics and Realism]: surface triangulation;

General Terms: Algorithms;

Additional Key Words and Phrases: surface construction, surface triangulation, planar contours;

1. Introduction

Many scientific and technical endeavors require the reconstruction of a three-dimensional solid from a collection of two-dimensional planar contours. These contours are obtained by some sensor method that samples the original three-dimensional solid along a finite number of parallel planes. The data extracted from that set of parallel planes are contours that lie along the solid's exterior and interior surfaces. The contours on the parallel planes appear as line segments. The

‡ This work has been supported by the NPS Foundation Research Program.

line segments are either closed loops, open segments, or single points. The goal of surface construction is the formation of surface patches between contours on adjacent planes such that an approximation of the original three-dimensional solid is formed.

Surface construction by the triangulation of two-dimensional contours is the procedure by which a pair of parallel, planar contours are "mapped together" and then "triangulated" into surface patches that form a surface display. The *mapping operation* of the surface construction algorithm identifies which contours on consecutive, parallel planes should be mapped together, and exactly which portions of those contours should be connected. The *triangulation operation* forms the connections between contours on adjacent planes by building triangular tiles between those mapped contours. Each triangular tile is built from an individual line segment from one contour and a single point from the end of a line segment on the other mapped contour. This tiling operation is performed for all line segments in the connect region of each mapped contour. The *connect region* is that section of coordinates designated as mappable for a pair of contours on consecutive planes.

Notationally, this problem has been specified as follows:

"An unknown three dimensional solid is intersected by a finite number of specified parallel planes. . . .

The only information about the solid consists of the intersections of its surface with the planes. Each of these intersections is assumed to be a simple closed curve. These curves are not completely specified; instead, a finite sequence of points encountered during a positive (counterclockwise) traversal of each of the original curves is given. The curve segment between two consecutive points is approximated by a linear segment, called a contour segment. . . .

We reduce the problem of constructing such an approximating surface to one of constructing a sequence of partial approximations, each of them connecting two contours lying on consecutive planes (Figure 1.1).

Let one contour be defined by the sequence of m distinct contour points $P_0, P_1, \dots, P_{(m-1)}$, and let the other contour be defined by the sequence of n distinct contour points $Q_0, Q_1, \dots, Q_{(n-1)}$. We note that P_0 follows $P_{(m-1)}$ and that Q_0 follows $Q_{(n-1)}$, and so indices of P are modulo m and indices of Q are modulo n . We wish to create a surface between the contours P and Q . The surface is constructed of triangular tiles between these two contours. The vertices of these tiles are contour points, with the vertices of each tile taken two from one sequence and one from the other. Thus, each tile is defined by a set of three distinct elements either of the form $\{P_i, P_k, Q_j\}$ or $\{Q_i, Q_k, P_j\}$ (Figure 1.2). . . .

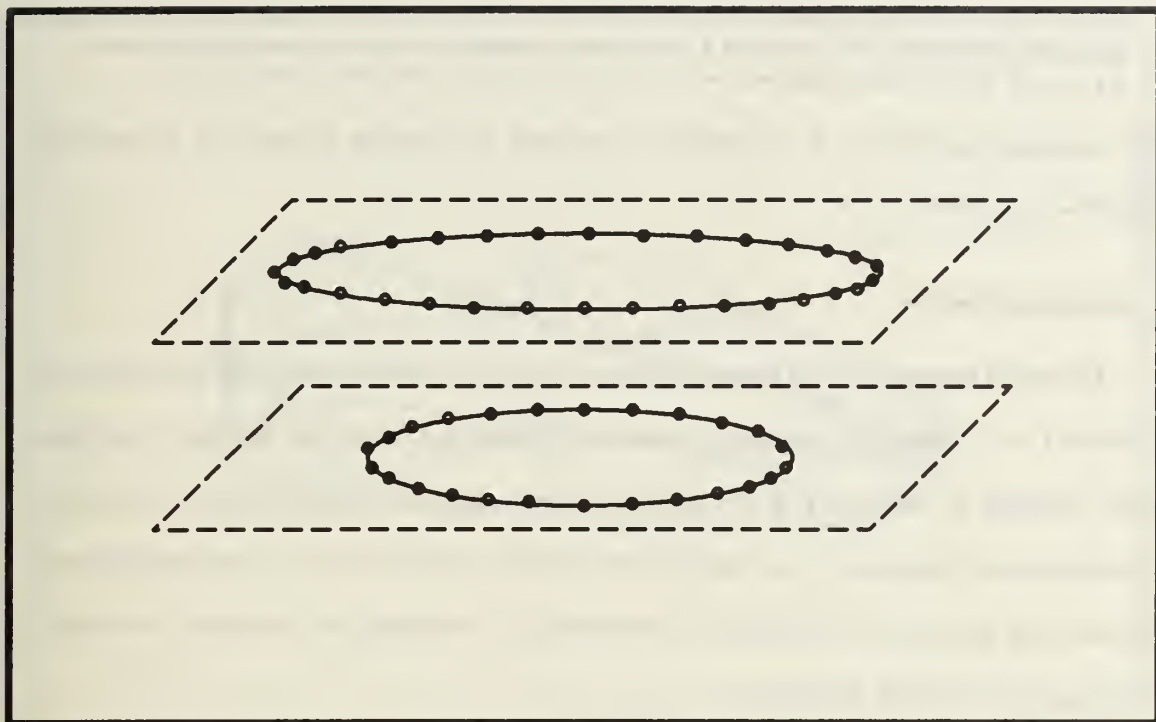


Fig 1.1 - Two contours on adjacent, parallel planes.

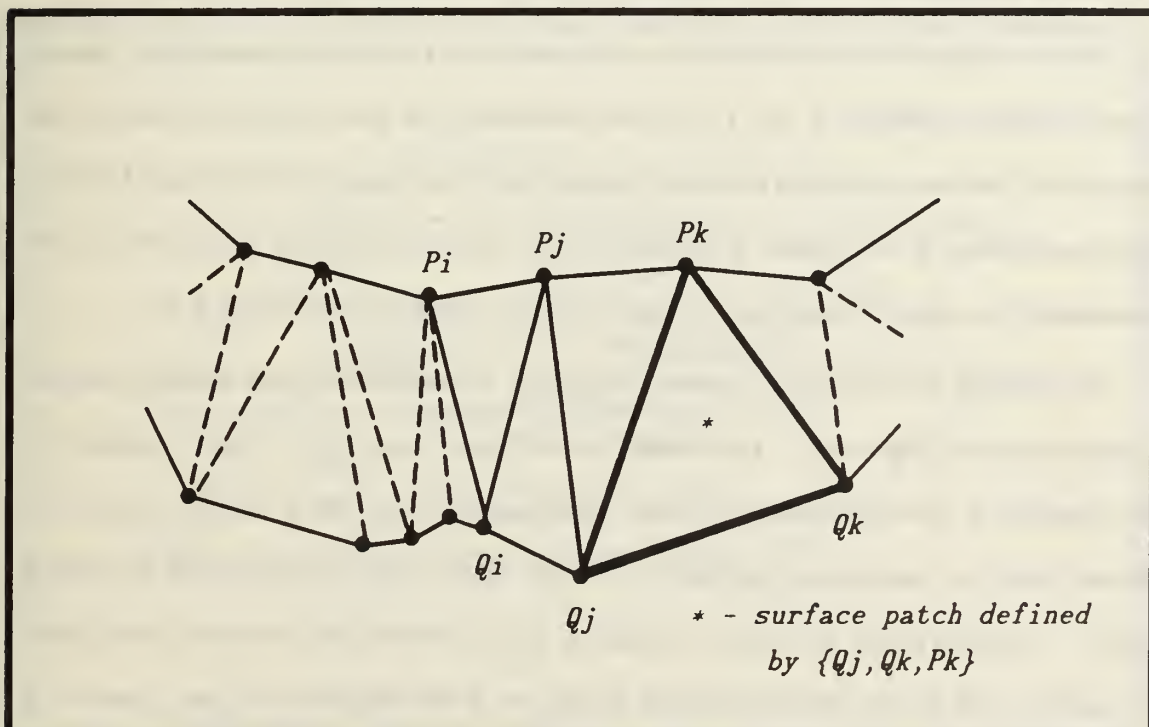


Fig 1.2 - Mapped connections into triangulated surface patches.

Each tile's boundary will consist of a single contour segment and two spans, each connecting an end of the contour segment with a common point on the other contour." [2]

This notational specification of the problem is consistent in all papers accessible in the literature on surface construction [1 - 4].

2. Literature Review

The initial emphasis of this paper is a review of the four previous algorithms for surface construction [1 - 4]. These four algorithms provide the background for the development of our algorithm. Included in this review is a discussion of each algorithm's capabilities and limitations. After this review, we present a new algorithm for surface construction that is more comprehensive than any that has previously appeared in the literature. Following that discussion, we examine the limitations of our new algorithm.

2.1. Fuchs Algorithm

The first algorithm we examine for the reconstruction of a three-dimensional object from its planar contours is presented in [2]. The problem statement from that article (reproduced in our introduction) has been used in all subsequent papers which build upon the Fuchs algorithm. The major contribution of that article, in addition to the concise statement of the problem, is the presentation of an algorithm capable of connecting simple, closed contours (Figure 2.1).

The problem with the Fuchs algorithm stems from its inability to handle multiple contours on adjacent planes (Figure 2.2). Additionally, no mechanism is provided to handle partial contour mappings or open (non-closed) contours. With respect to the case of multiple contours on adjacent planes, no mechanism is provided to identify which of the contours should be mapped together. The general case for surface construction is to have multiple contours on each plane. The problem with partial contour mappings is that the Fuchs algorithm can only construct a complete triangulation between adjacent contours. This limitation disallows partial triangulations of contours. Such partial mappings often are indicated for cases of dissimilarly sized contours. Finally, the problem of open contours can be attributed to algorithm generality. A mechanism that solves the partial contour mapping problem can also solve this problem.

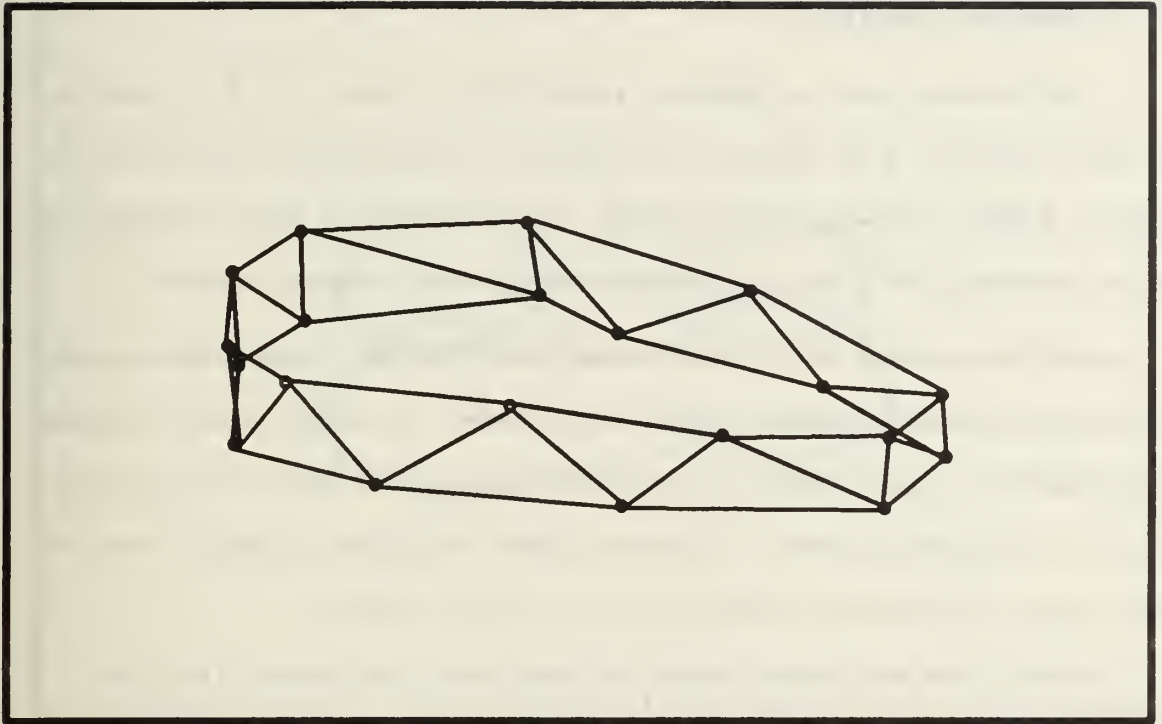


Fig. 2.1 - Triangulated pair of simple, closed contours.

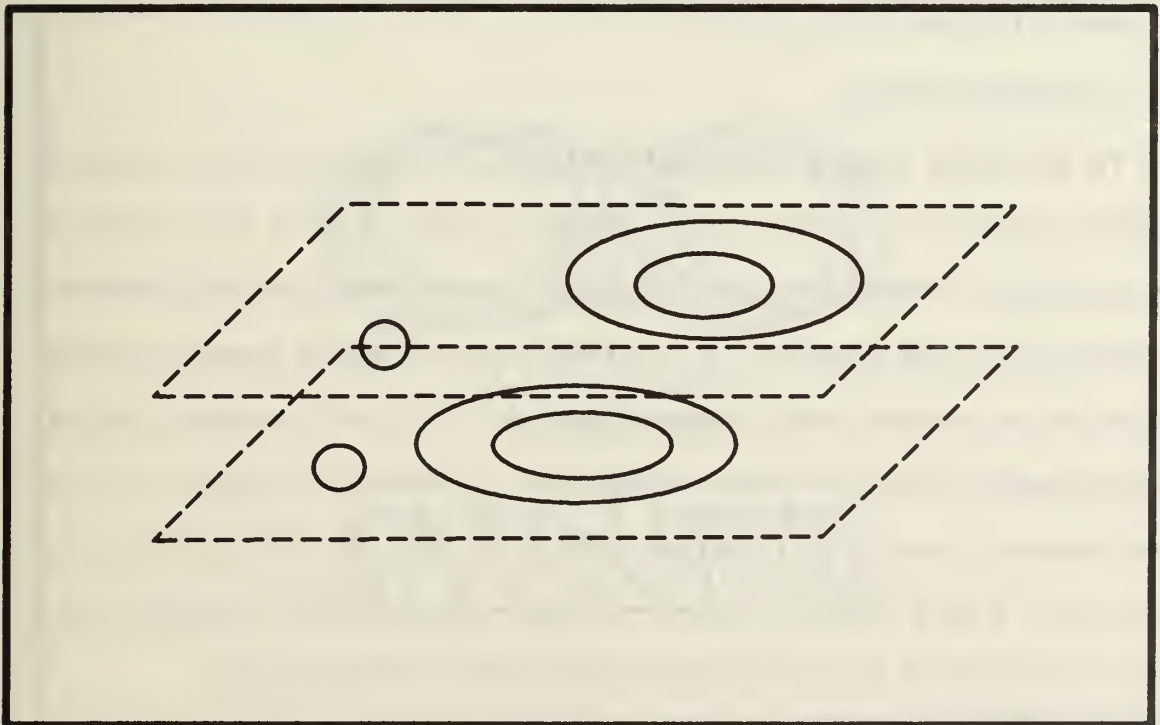


Fig. 2.2 - Example of multiple contours per plane.

2.2. Christiansen Algorithm

In the Christiansen paper, an algorithm is presented which is similar to the Fuchs algorithm. The major dissimilarity is the inclusion of a mechanism to facilitate human interaction for the resolution of highly ambiguous contour mappings. Human interaction is used to determine the relative connection points in the contour mapping process for highly convoluted contours.

As the Fuchs algorithm can, this algorithm can handle mappings of simple, closed contours. It also has capabilities for mapping together simple branches. An example of such branching, seen in Figure 2.3, is a pair of contours on one plane being mapped to a single contour on an adjacent plane. This capability allows the algorithm to handle simple cases of multiple contours on adjacent planes. The method by which this problem is solved is as follows:

1. Introduce a new node midway between the closest nodes on the branches. The Z coordinate of this node is the average of the Z coordinates of the two contour levels (planes) involved.
2. Renumber the nodes of the branches and the new nodes such that they can be considered as being one loop (Figure 2.4).
3. Triangulate as usual [1].

The Christiansen algorithm is not capable of handling open contours, nor is it capable of handling complex cases of multiple contours on adjacent planes, except by way of expensive human interaction. A final note of interest with respect to this algorithm is the use of a heuristic for selection of the nodal connections. In cases where contours on adjacent planes are mutually centered and are reasonably similar in size and shape, selection for nodal connection is based on "shortest diagonal" rather than minimum triangular area [1]. During this operation, one of two nodes is selected to create the next triangular surface patch. The nodes under consideration are the two "next" nodes of each contour. By determining the length of each of the possible diagonals for the surface patch, the connection node is selected based on minimum length.

2.3. Shantz Algorithm

The algorithm presented in [4] extends the algorithms of Fuchs and Christiansen to handle contour defined objects which are highly branched and have holes. Multiple contours on adjacent

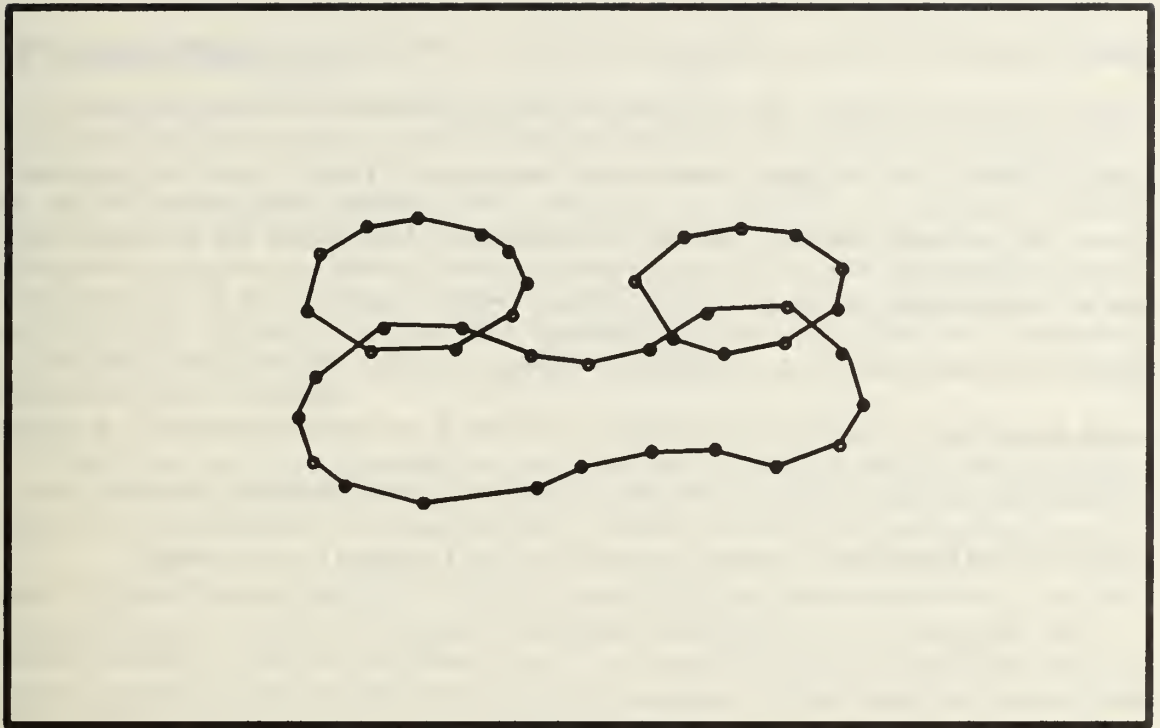


Fig. 2.3 - Simple case of branching.

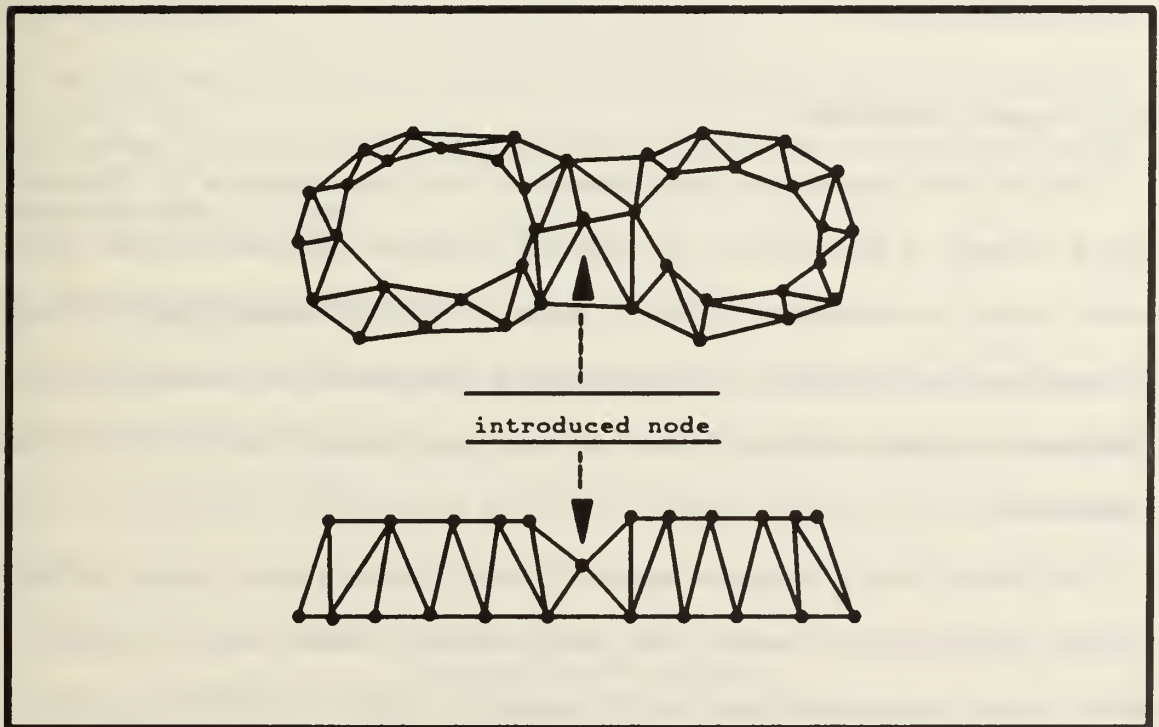


Fig. 2.4 - Triangulation scheme for branching.

planes are handled by "first concatenating the contours on each plane into a single large contour using minimum distance links, then performing the mapping between the resulting composite contours [4]." Shantz uses the simple, closed contour mechanism of Fuchs to form the connections between the composite contours. Once the connections have been formed, the extraneous ones (due to concatenation) are removed. Some difficult multiple contour cases for this algorithm require human interaction to solve ambiguities. Shantz states that this should be avoided since human interaction is "extremely labor intensive." He cites a case which required 50 to 80 hours of contour splitting, using an interactive cursor, to produce a surface display for the highly convoluted cortex and basal ganglia contours (extracted from the Livingston brain database).

This algorithm, as is the Christiansen algorithm, is limited in its ability to handle cases of open contours and partial contour mappings. Also, cases of multiple contours on adjacent planes can be handled only when a composite contour can be formed, or when ambiguities are resolved via human interaction.

2.4. Ganapathy Algorithm

The most recent algorithm for surface construction from planar contours is [3]. That algorithm is essentially an improvement on the Fuchs and Christiansen algorithms for simple, closed contours, without the capabilities described by Shantz. Like Fuchs, Ganapathy assumes a complete mapping of contours, which is not always possible. The improvement over the Fuchs and Christiansen algorithms is attributed to the use of a more computationally expedient heuristic for triangulations.

The problem with the Ganapathy algorithm is that it presents a general solution for handling only the simple case of mapping single, closed contours on adjacent planes. The issues of multiple contour mappings and partial contour mappings are ignored. Additionally, no mechanism for user interaction is provided for resolving mapping ambiguities, further limiting the algorithm to simple cases.

2.5. Literature Deficiencies

None of the above papers provides a complete solution to the problem of surface construction via the triangulation of contours. What is required is an algorithm with capabilities for multiple contours per plane and partial contour mappings. Additionally, the algorithm should support simple cases of branching and provide a mechanism for human interaction for the resolution of highly ambiguous mappings.

The surface construction algorithm we present handles not only the simple contour mapping problem, but also provides a more comprehensive procedure for solving the multiple contours per plane and partial mapping problems. The only capability lacking from our algorithm is that for handling branching as per the Christiansen paper. A discussion of our algorithm follows, with a proposed solution for handling cases involving branching.

3. The Algorithm

We begin the presentation of our algorithm by first discussing the known input and output data structures. Following that section, an overview of the major parts of the algorithm precedes a detailed discussion of the parts.

3.1. Input/Output Specifications

The problem of surface construction of an object from a set of planar contours, as seen in Figure 3.1, can be reduced to one of constructing the surface triangulations between two adjacent planes. The specification of the problem can be best viewed by detailing the known input data structures:

<code>total(i):</code>	number of contours on plane i.
<code>start(j,i):</code>	start of contour j on plane i.
<code>length(j,i):</code>	number of coordinates in contour j on plane i.

`type(j,i):` type of contour j on plane i (CLOSED_LOOP, OPEN_SEGMENT, or SINGLE_POINT).

`interior(j,i):` value of contour j's interior with respect to the contour line (HIGH, LOW, or INDETERMINATE).

`coords(XYZ,pointer,i):` input coordinates for all contours on plane i. To isolate contour j on plane i: We run (pointer = start(j,i) + k - 1), where k = 1, length(j,i).

From the above data, we desire to produce the following output data structures:

`num_coords:` number of coordinates generated for the two input planes.

`new_coords(XYZ,num_coords):` coordinates generated by the surface construction process for the two planes.

`new_conns(num_coords):` drawing instructions for each coordinates generated (SETPOINT, DRAWTO, DRAWPOINT).

If the output data is in the form of triangular surface patches, an alternative data structure is required:

`num_patches:` number of surface patches generated for the input two planes.

`new_coords(XYZ):` new coordinates generated by the connection process.

`patches(3,num_patches):` a 3 by num_patches array of triangles.

3.2. The Algorithm

Our surface construction algorithm is composed of the following six outlined steps:

(1) *Input and Inventory Compilation:* The data structures defining the contours are processed to extract the pertinent data. This data includes the number of contours per plane, the coordinates defining these contours and the types of the contours. Additionally, two-dimensional bounding boxes are described about each contour for processing consideration in step 2. This compilation of

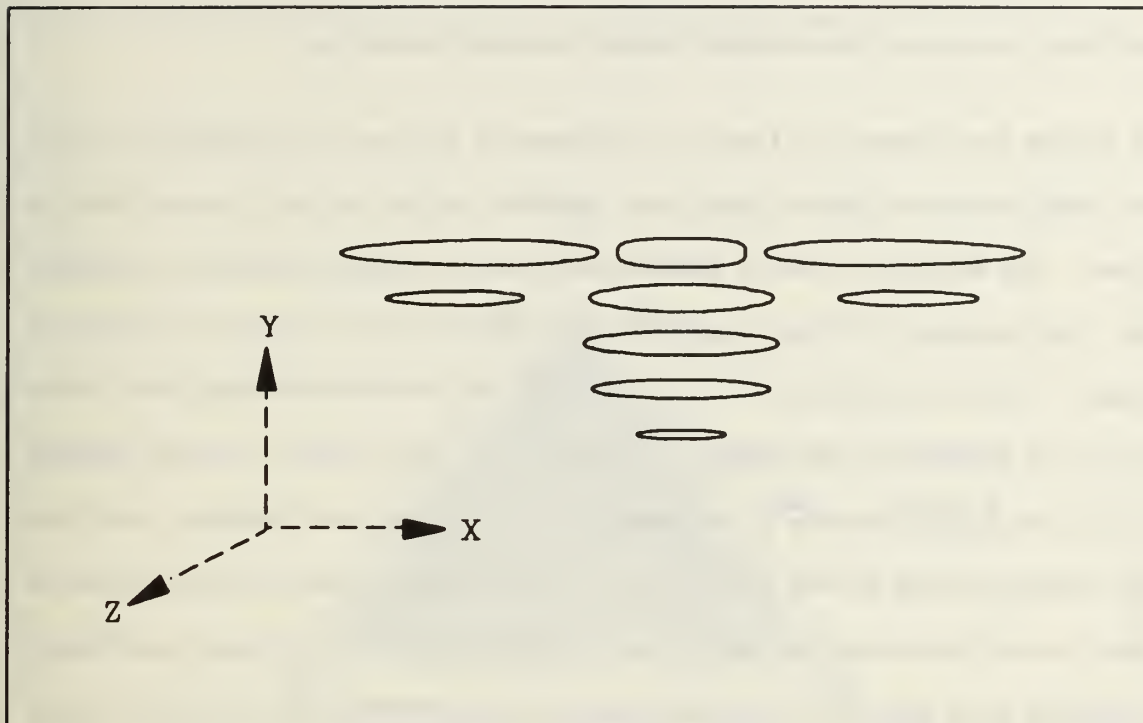


Fig. 3.1 - A partial set of planar contours from a 3D Z^2 -orbital of a hydrogen molecule.

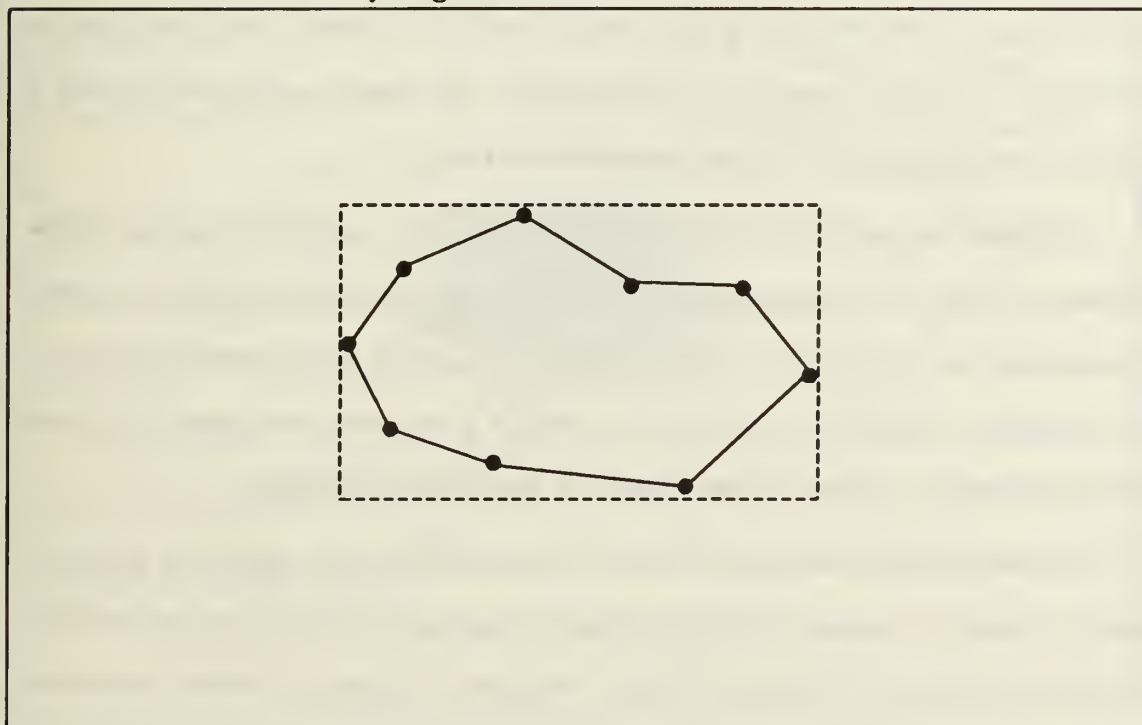


Fig. 3.2 - Two dimensional bounding box used for determining overlap percentage value.

data is used to create the data structures required for surface construction.

(2) *Overlap Determination and Contour Item Mapping:* In this step of the algorithm, we determine which contours on adjacent planes have significant overlap, and which contours' exteriors are near. This information is used to designate which contours should be connected via triangulations. The assignment of overlap is accomplished through the use of a value for the overlap percentage. This value is computed from the areas of the two-dimensional bounding boxes of each contour (see Figure 3.2). The overlap percentage is used to give priority to contour mappings that have the highest percentage of total overlap area. In this step of the algorithm, we also perform consistency checks for each contour pair. One such consistency check is executed using the contour interior specification and the overlap percentage value. Contour interior specifications are assigned as the value of a contour with respect to its immediate interior. As such, a contour is LOW valued if it is taken from the exterior of a solid object, such as the skin of an apple. Conversely, a contour is HIGH valued if its immediate interior is non-solid. Using these pieces of information, we are able to eliminate contour mappings of high overlap percentage which result in erroneous approximations of the original three-dimensional solid.

To illustrate the application of this consistency check, let us consider the mapping example of Figure 3.3. Here we are presented with a set of contours taken from a solid cone standing within a hollow cone. In this case, contour 1 on plane 1 has a high overlap percentage with contour 2 on plane 2. However, since contour 2 on plane 2 is low valued with respect to its solid interior and contour 1 on plane 1 is high valued, this mapping can be eliminated.

The interior specifications are also used to determine whether the mapping is interior to interior or exterior to exterior. An interior to interior mapping is one which maps the interior of one contour to the interior of another contour. This form of mapping is indicative of contours taken from a surface with a shallow gradient, i. e. - a surface where the mapped contours are of similar size and shape, and where the contours have significant overlap. An exterior to exterior mapping is one which maps the exterior of one contour to the exterior of another contour. This form of mapping is indicative of contours taken from a surface with a steep gradient, i. e. - a

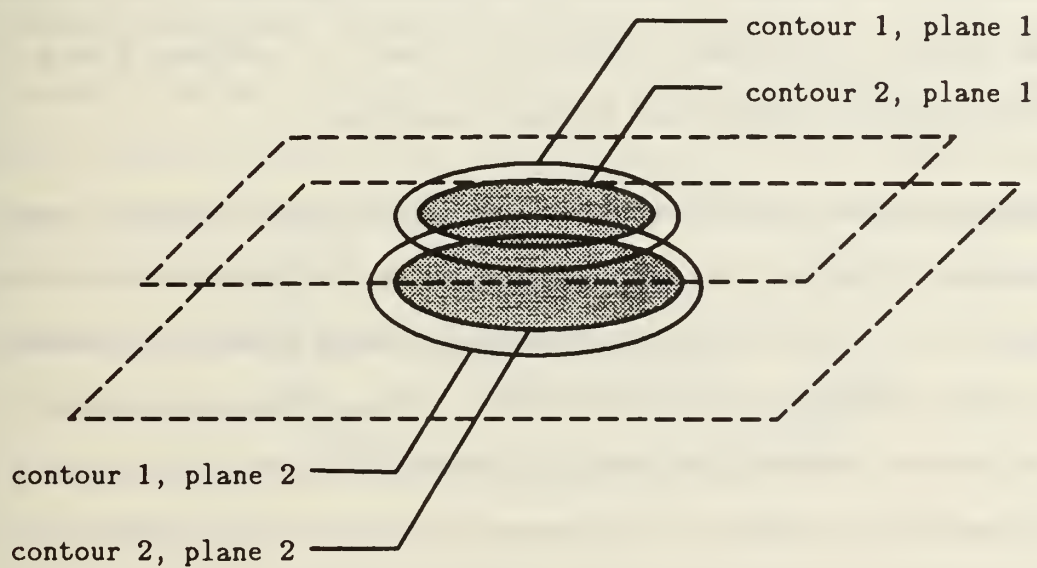
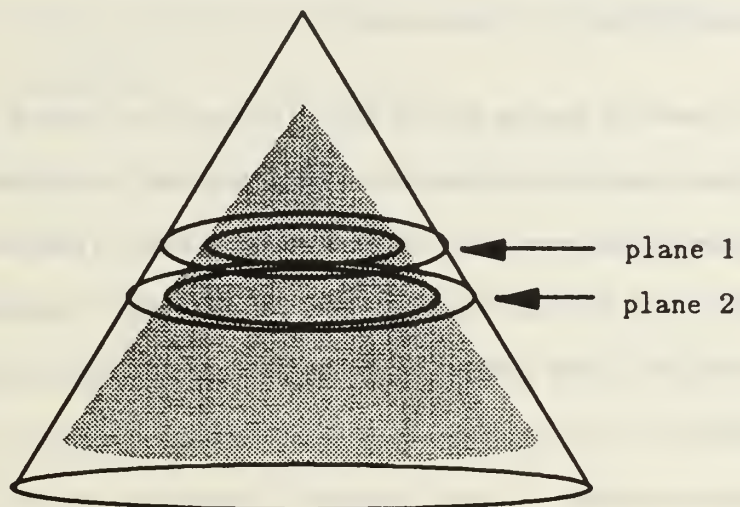


Fig. 3.3 - Example of consistency check using item interior specifications with overlap percentage values.

surface where mapped contours are of dissimilar size and shape, and where the contours' overlap percentage is slight. Interior to interior mappings are more common. The exterior to exterior mapping is indicated for cases of two contours with a low percentage of overlap and differing interior specifications (HIGH:LOW, or vice versa).

(3) *Form the Coordinate Mapping for each Mapped Contour Pair:* For each coordinate pair from step two, we form a complete coordinate to coordinate mapping. A coordinate mapping is a tentative set of triangulation connections between the contour pairs. There are two procedures for determining this initial coordinate mapping. The procedure used is dependent on the type of mapping found for the paired contours in the previous step (interior to interior, or exterior to exterior). Additionally, both procedures try to form triangulation segments of shortest length, as in the Christiansen algorithm. A general statement of this selection process is that we are trying to map coordinate i of contour n , plane 1 to coordinate j of contour m , plane 2 such that the distance between the two coordinates is minimized. An additional qualification to this distance minimizing criterion is that coordinate connections do not cross, i. e. - coordinates 3 and 4 of plane 1 are not mapped to coordinates 6 and 5 of plane 2 respectively.

(4) *Continuity Recognition:* The coordinate to coordinate mapping formed in step three is examined for continuity. Continuity, in this case, is defined as follows. First, we form continuous sets of coordinates from the coordinate mapping such that each coordinate of each set is constrained within a coordinate tolerance and within a distance range. The coordinate tolerance factor is a ratio of the number of coordinates in the larger contour divided by the number of coordinates in the smaller contour times a window value. The tolerance factor is used to group coordinates into a single set based upon their mapped coordinate number being within plus or minus tolerance of the last mapped coordinate added to the set. The tolerance sets formed are then compared for overlapping distance ranges. Any sets that have overlapping distance ranges are then merged. The merged set with the smallest distance in it is the set of coordinates for which connections should be generated. All other coordinates are left unconnected.

(5) *Mapping Cancellation*: Once we have decided to generate the connections for a part of a contour, we cancel any further mappings to that piece of the contour. This operation is required for partial mappings in which two or more contours on one plane are to be mapped to a single contour on another plane. Also, this cancellation precludes connecting contour points which have already been selected for connection.

(6) *Connection Formation*: We generate the coordinates for the triangulation connections specified in step four. "In between" coordinates, coordinates not directly mapped but within the tolerance factor for the connection mapping, are also added to the picture. The goal of the process is to form minimum area triangular surface patches for each segment of the mapped connection region.

3.2.1. Input and Inventory Compilation

The input data to the algorithm consists of the contour descriptions for two adjacent planes of a three-dimensional solid. The purpose of this step of the algorithm is to segment this data into separate contour descriptions and to determine the individual characteristics of each contour. Figure 3.4 consists of two adjacent planes, each having three concentric rings of similar shape and continuity. Figure 3.6 consists of two closed loops on each of its planes. Plane 1 has two small interior lobes, while plane 2 has one large surrounding contour with a small interior contour. The contour descriptions for these figures are composed of:

- the starting coordinate location,
- the total number of coordinates,
- the contour types,
- the interior values, and
- the contours' two-dimensional bounding boxes.

With the exception of the interior values, all of these characteristics are easily obtainable from the input data.

The procedure necessary to obtain the contour interior specifications requires an evaluation

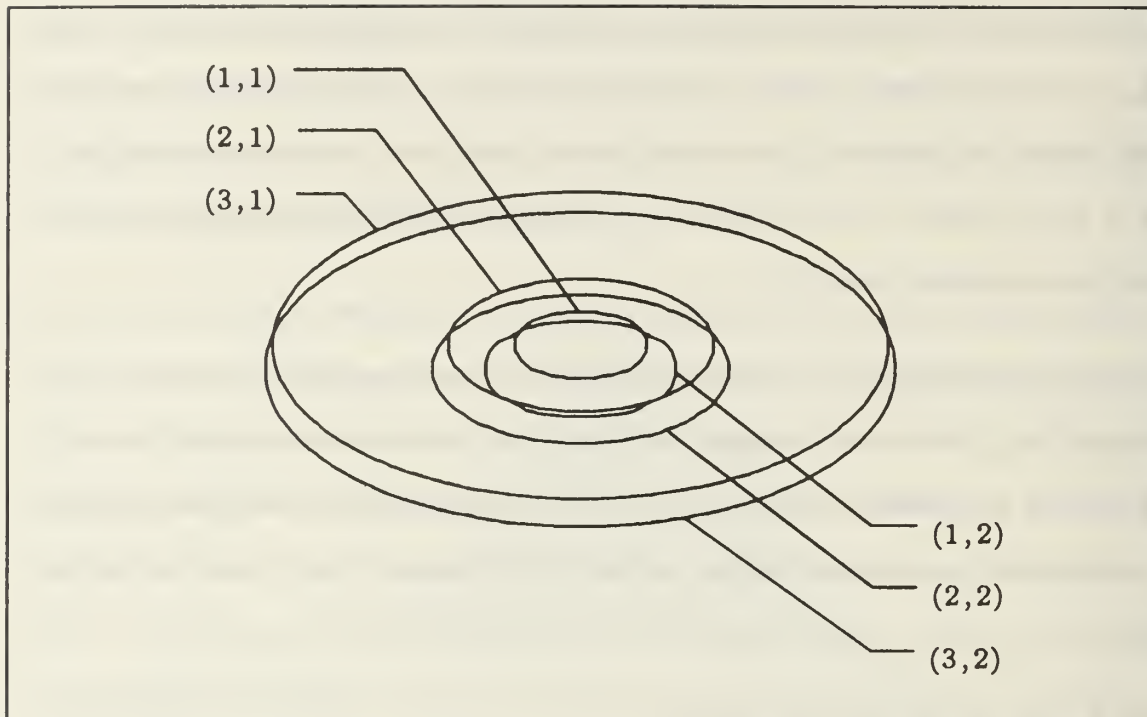


Fig. 3.4 - Example of multiple contours per plane on adjacent planes.



Fig. 3.5 - Connection of Figure 3.4.

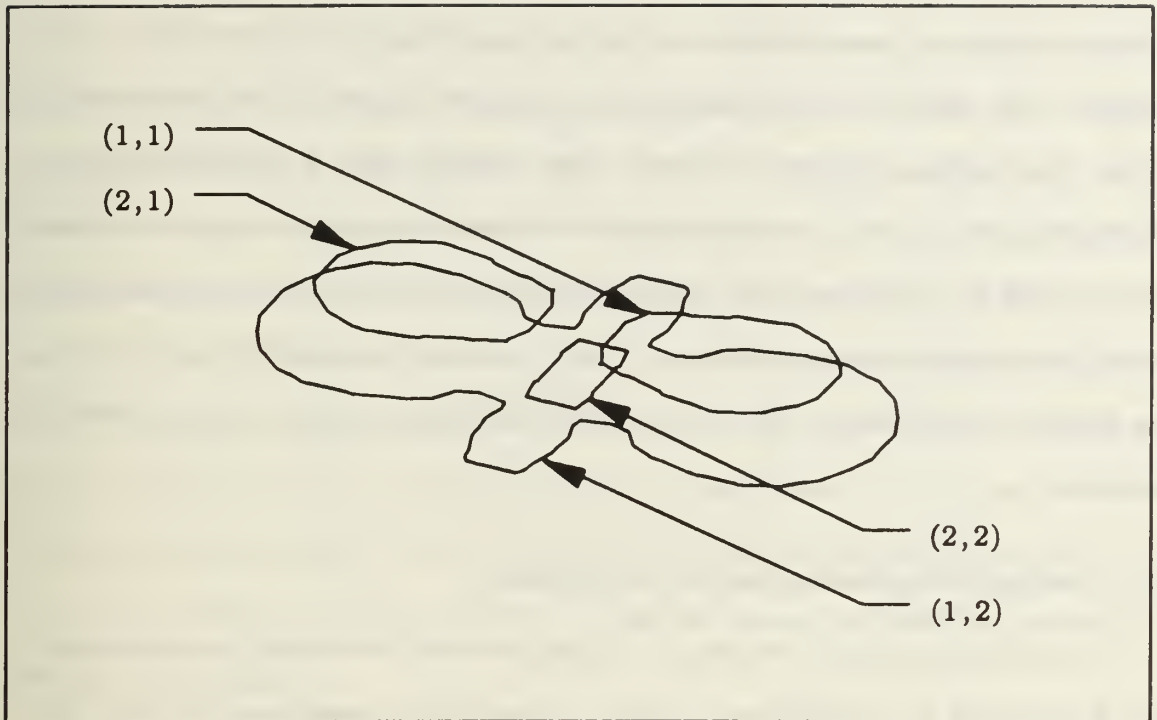


Fig. 3.6 - Example of a set of contours requiring partial mappings and an exterior to exterior mapping; (1,1) and (2,1) to (2,2).

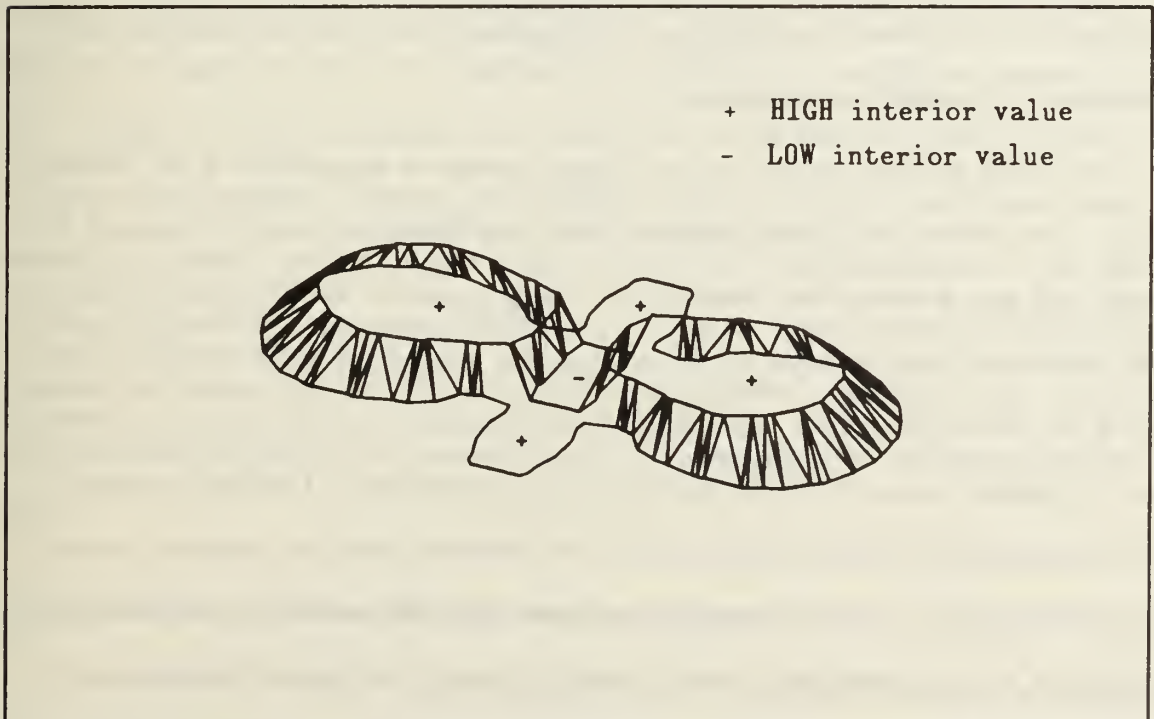


Fig. 3.7 - Connection of Figure 3.6, with contour interior values for each contour.

of the data values lying along and interior to the contour (see Figure 3.3). If these values are not contained in the input data, a mechanism needs to be provided to allow for user specification of contour interior values. The range of interior values is HIGH, LOW or INDETERMINATE. Without this value, the contour pairing operation encountered in the multiple contours per plane situation is difficult. In that case, some form of human interaction is necessary to designate which pairs of contours should be mapped together. If an interior value is not available, and the mapping situation is not complex, it can be set to INDETERMINATE without surface construction degradation.

3.2.2. Overlap Determination and Contour Mapping

The overlap determination and contour mapping procedure of the surface construction algorithm is the process by which tentative contour to contour mapping assignments are made. The contour characteristics which are necessary for this procedure are the two-dimensional bounding boxes and the contour interior specifications. This mapping process is the key component in the disambiguation of multiply paired contours.

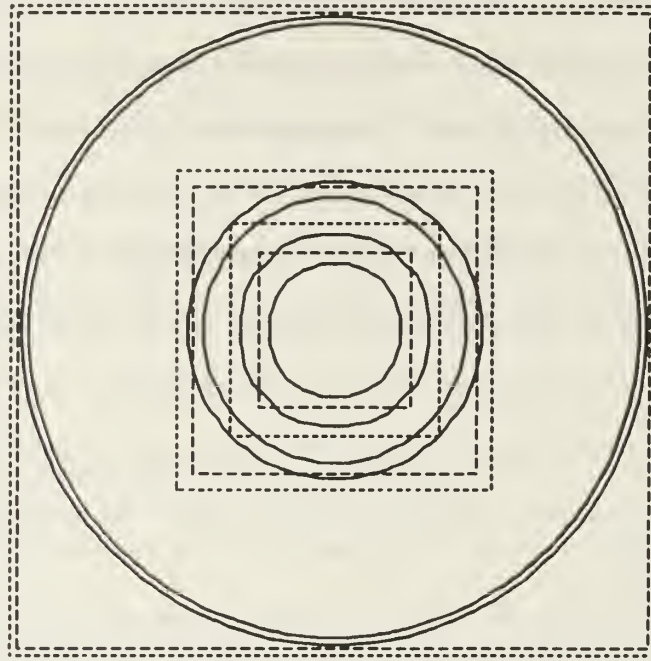
The overlap determination and contour mapping procedure is accomplished in the following manner. First, the two-dimensional bounding box of each contour on plane 1 is compared for overlap with the two-dimensional bounding box of each contour on plane 2. The coordinates which define these bounding boxes are the minimum and maximum X and Y coordinates from each of the contour descriptions. (Additionally, these coordinates are adjusted by a constant value to promote overlap for exterior to exterior mapping situations.) From this operation, a table called the overlap table is produced. It is a two-dimensional table that contains a value for each possible pairing of contours between the two planes. The value recorded in each table entry indicates the extent to which each contour overlaps. If there is no bounding box overlap for a pair of contours, a value of 0.0 is recorded in the table. If there is overlap, the value recorded in the table represents the percentage of overlap with the larger of the two contours. This value is computed by dividing the area of the bounding box overlap by the area of the bounding box of the larger contour.

After the overlap percentage has been computed for a contour pairing, it is used in conjunction with the interior specifications to determine the mapping type for the contour pair. An interior to interior mapping is indicated when a high percentage of overlap (greater than 10%) exists for a pair of contours. A consistency check for matching interior specifications is performed for every pair of contours that exhibits this high an overlap. The consistency check requires that each contour pair have either HIGH:HIGH, LOW:LOW, or INDETERMINATE:anything (HIGH or LOW) interiors. Contour pairings with high overlap but inconsistent interior specifications result in an adjustment to the overlap table of 0.0 percentage of overlap. An exterior to exterior mapping is indicated when the overlap percentage is low (less than 10%) and item interiors are non-matching. Finally, all contours with low overlap percentages and matching interiors are zeroed in the overlap table.

Figures 3.8 and 3.9 graphically represent the overlap determination and contour mapping for Figures 3.4 and 3.6. Included in these figures are the overlap tables produced by this procedure. The table in Figure 3.8 shows three valid overlap percentages for three different contour pairs: (1,1) - (1,2), (2,1) - (2,2), and (3,1) - (3,2). Four of the entries have been zeroed by the consistency check mechanism. Without this capability, high valued overlap percentages would appear in the overlap table with human interaction required for their disambiguation. The table in Figure 3.9 shows two high overlap percentages and two low overlap percentages. This data indicates that contours (1,1) and (2,1) both map interior to interior with contour (1,2). The low overlap percentages indicate that contours (1,1) and (2,1) map exterior to exterior with contour (2,2).

3.2.3. Form the Coordinate Mapping: Interior to Interior

The coordinate mapping formation procedure for each coordinate pair having a non-zero overlap (in the overlap table) begins with the pair having the largest overlap percentage. All remaining steps in the surface construction algorithm are carried out on this pair before the next pair of contours is considered for mapping. The operation for mapping paired contours is carried out in a largest to smallest overlap percentage order. Since exterior to exterior mappings are

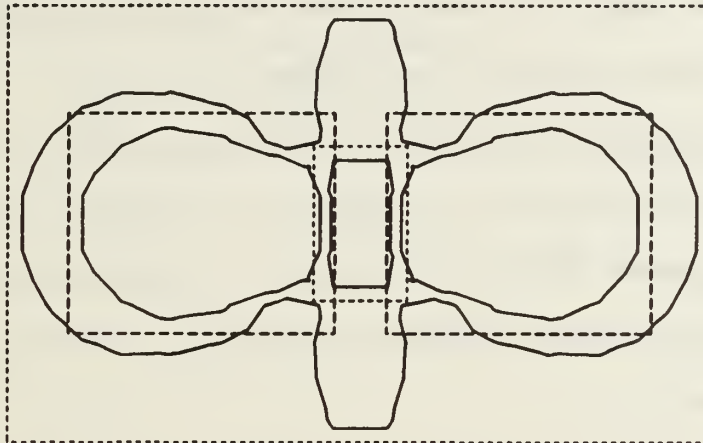


OVERLAP TABLE

Plane 2

		CONTOUR 1	CONTOUR 2	CONTOUR 3
Plane 1	CONTOUR 1	95.6916	0.0	11.1493
	CONTOUR 2	0.0	81.3006	0.0
	CONTOUR 3	0.0	0.0	52.4872

Fig. 3.8 - Bounding boxes and overlap table produced for Figure 3.4



OVERLAP TABLE

Plane 2

		CONTOUR 1	CONTOUR 2
Plane 1	CONTOUR 1	19.0295	5.4386
	CONTOUR 2	19.0295	5.4386

Fig. 3.9 - Bounding boxes and overlap table produced for Figure 3.6

indicated only in situations where the overlap percentage is low, they are considered for mapping only after all interior to interior mappings have been performed. This study follows that ordering and completes the description of the interior to interior mapping process before considering the separate process necessary for exterior to exterior mappings.

The first operation performed on an interior to interior overlap pair is the determination of which contour is interior to the other. This assignment is accomplished by comparing bounding box areas for the contour pair and designating the contour which has the smallest area as interior. Once the interior contour assignment has been made, the center coordinate of that contour's bounding box is computed.

The knowledge of the center coordinate of the interior contour is used in the following manner. For each coordinate of the inner contour, we determine which coordinate of the outer contour is closest to a vector drawn from the center coordinate of the inner contour through the coordinate of the inner contour (see Figure 3.10). We add the qualification that the outer coordinate selected by this procedure must be farther from the center coordinate than the inner coordinate. Also, the outer coordinate must be on the same side of the vector as the inner coordinate. The outer coordinate selected by this mapping process is recorded as the tentative coordinate map coordinate for each inner coordinate. We also record the two-dimensional distance from each inner coordinate to its tentatively mapped outer coordinate. The resulting data structure contains the mapped outer coordinates with their companion distances.

The tentative connection map for Figure 3.4 is very good. Due to the similarity in size and shape of the mapped contour pairs, there is very little variation in the mapped distance values and the coordinates selected for mapping appear sequential. On the other hand, it can be seen in Figure 3.11, that large variations in distance values result from this tentative mapping process, and mapped outer coordinates appear with large gaps in the sequencing. This is due to the dissimilarity of the contour pair; the inner contour is relatively simple and much smaller than the convoluted outer contour. The procedure used to delineate a correct mapping from this tentative mapping is described below.

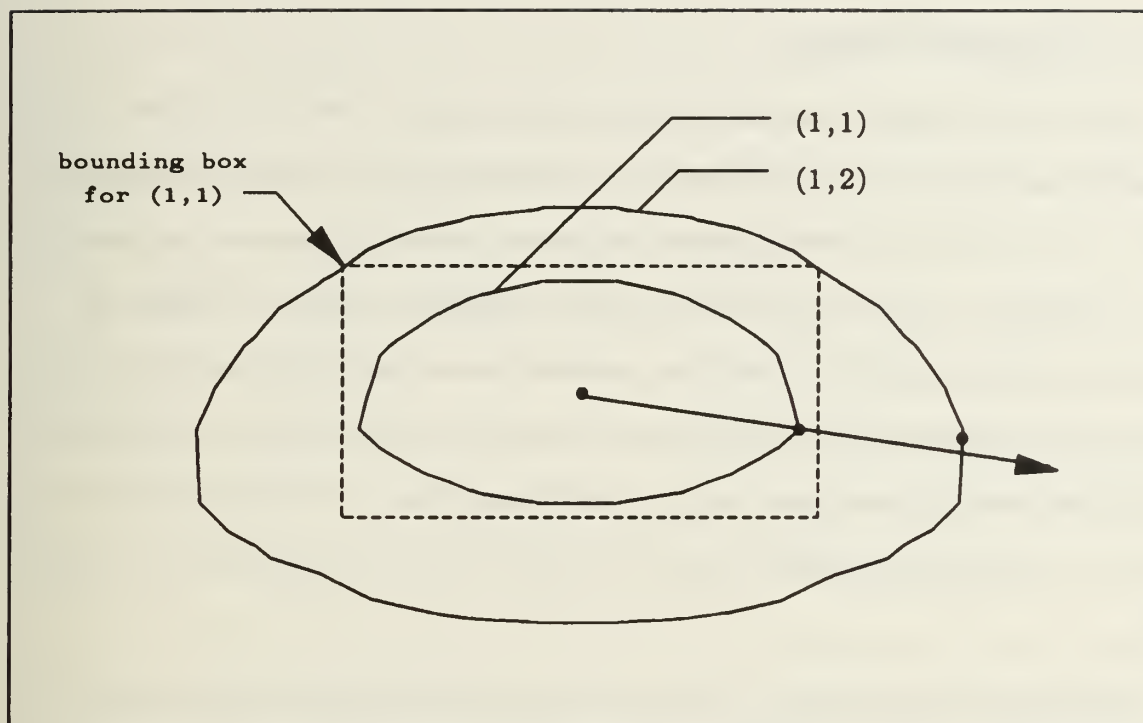


Fig. 3.10 - Vector radiating from center coordinate through the interior coordinate towards the outer contour for tentative mapping

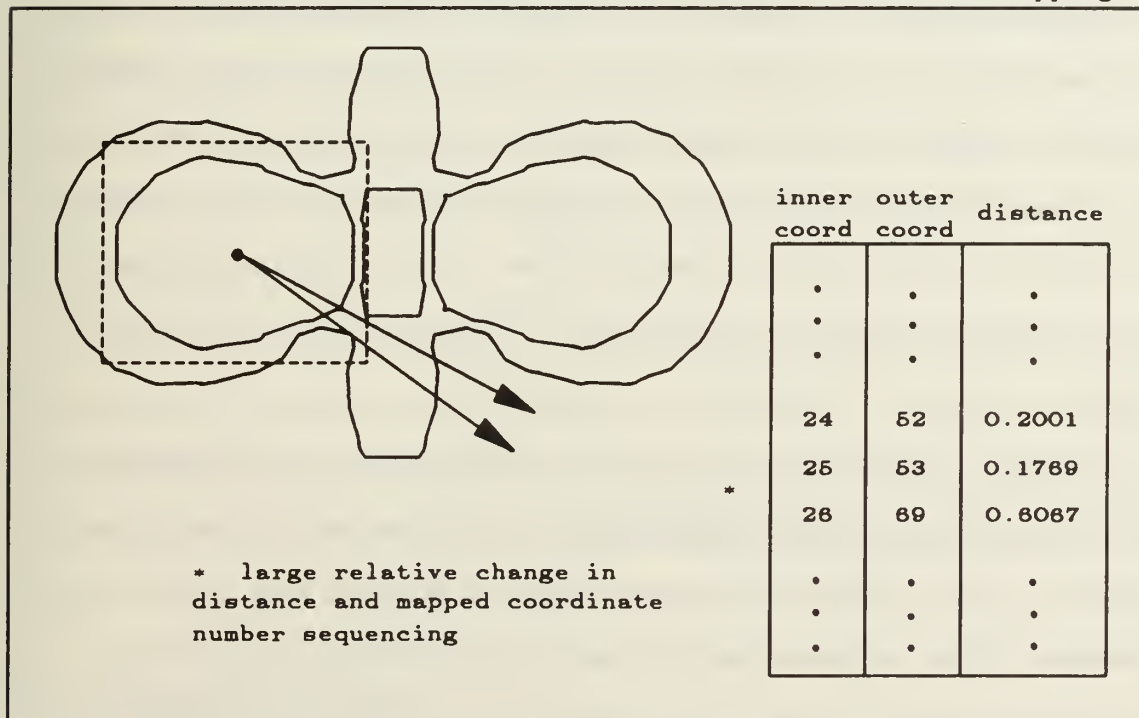


Fig. 3.11 - Example of a case where tentative mapping coordinates and associated distances vary greatly.

3.2.3.1. Continuity Recognition

The continuity recognition procedure uses the tentative connection map and associated distances for a pair of contours to determine the set of coordinate mappings that should be made for that pair. In the previous step of the algorithm, we produced the tentative connection map for all of the coordinates of the inner contour. This provides a rough approximation of the final mapping, but it must be noted that all of the inner coordinates may not necessarily be involved in the final mapping for that pair. The continuity recognition procedure builds sets of coordinate mappings that are both continuous and of similar mapped distance range. These continuity sets are then used to determine the coordinate sequences that should comprise the final connection mapping.

The first step in this procedure is to assign each coordinate pairing of the tentative connection map to an initial continuity set. This is accomplished by stepping through the coordinates of the inner contour in sequence and comparing each coordinate's mapped outer coordinate to the last coordinate added to the last created continuity set. If that coordinate is within a tolerance factor of the last coordinate added, it is added to that set. If the coordinate in question is not within tolerance, a new set is created with that coordinate mapping as its start. The tolerance factor used is a ratio of the number of coordinates in the outer contour divided by the number of coordinates in the inner contour times a window value. (The window value is discussed later in this paper.)

To illustrate this continuity set assignment, let us refer to the example in Figure 3.11. Here, the tolerance factor is 10 and the last coordinate considered was inner coordinate number 24. The next coordinate considered is coordinate 25, which is mapped to outer coordinate 53. This coordinate is within the tolerance factor of 10 and is added to the last created continuity set. Inner coordinate number 26 is mapped to outer coordinate 69. This outer coordinate is well outside of tolerance with the last coordinate added and therefore, a new continuity set is created with this coordinate mapping as its start.

This initial step of the continuity recognition process is a fast method for aggregating coordinate map pairs. In addition to building the initial continuity sets for the tentative mapping, we keep track of the minimum and maximum mapped distances for each continuity set. These values are used for merging continuity sets in the next step of the process.

The initial sets generated for Figures 3.4 and 3.6 are of particular interest. This step of the continuity procedure placed all of the tentative mappings for the coordinate mapping pairs for Figure 3.4 into a single set. This can be attributed once again to the contours' similar shapes and sizes. On the other hand, coordinate mapping pairs for the mapping (1,1) - (1,2) of Figure 3.6 resulted in 5 initial continuity sets with varying distance ranges (see Figure 3.12).

Once the initial continuity sets have been created for a contour pairing, we merge any sets that have overlapping mapped distance ranges. This merge process reduces the total number of sets and further aggregates the coordinate pair mappings to sets with coordinate number continuity and distance range similarity. In reference to our examples, no continuity set merge was required for Figure 3.4 due to its singular initial continuity set. Figure 3.12 shows the initial sets with distance ranges and the merged sets with distance ranges for the contour pairing (1,1) - (1,2) of Figure 3.6. In that figure, the 5 initial continuity sets have been merged into 3 sets of non-overlapping distance range.

After we have merged continuity sets, we need to determine which of those sets of coordinates mappings is the one that should be used for connection formation. The choice is clearly the set with the smallest distance range. With this decision, we validate all coordinate pairings that are members of this smallest distance set, and cancel all other coordinate pairings for that set of contours.

3.2.3.2. Mapping Cancellation

The validated coordinate connection map for the contour pair has significance beyond indicating which coordinates need to have connection segments generated. It also indicates "filled" connection positions. By filled we mean that once we have formed connections to a coordinate

Total Initial Sets = 5

Total Merged Sets = 3

Set Name	Min. Dist.	Max. Dist.
1	0.0176	0.1052
2	0.1769	0.2083
3	0.6067	0.6482
4	0.1769	0.2083
5	0.0176	0.0688

Set Name	Min. Dist.	Max. Dist.
1	0.0176	0.1052
2	0.1769	0.2083
3	0.6067	0.6482

Fig. 3.12 - Initial continuity sets and merged continuity sets for the contour pair (1,1) - (1,2) of Figure 3.6.

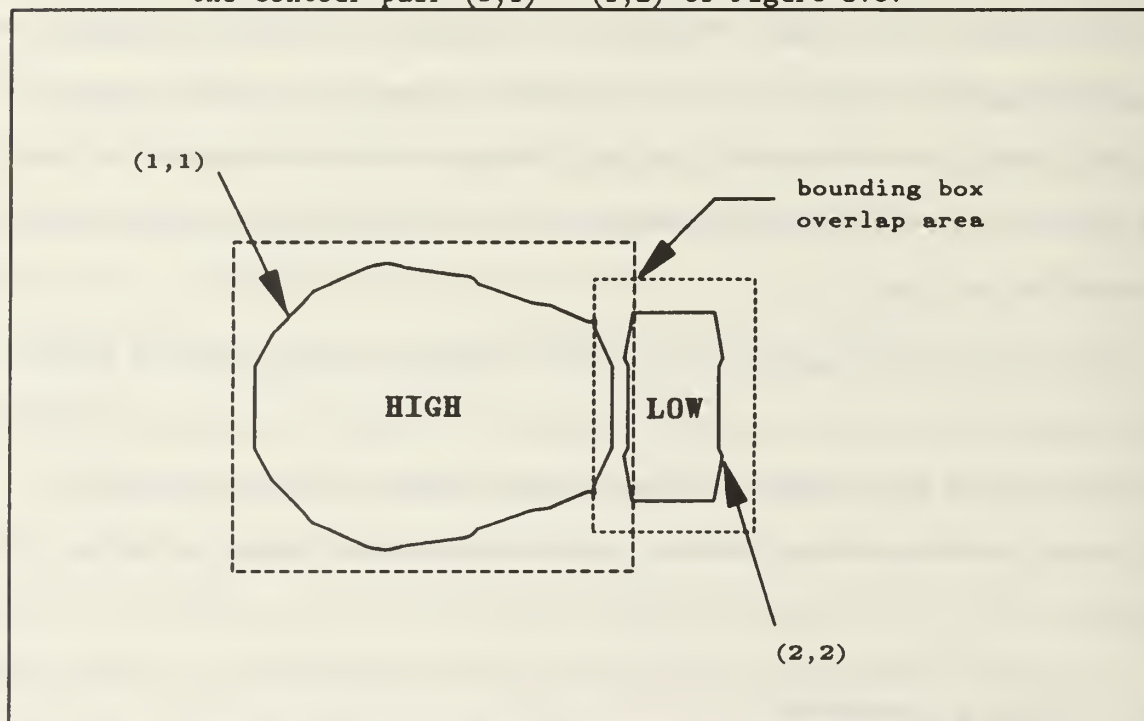


Fig. 3.13 - Bounding box overlap for exterior to exterior mapping. Only the coordinates within the overlap area are mapped.

segment of a contour, that segment should not be reused for any further mapping that occurs for the two current, adjacent planes. This mapping is both checked and recorded at this stage of the algorithm. Mapping cancellation examines the coordinate mappings for which a validated mapping has been assigned. If either of the two coordinates, inner or outer, has been assigned to a higher priority mapping for this pair of planes, then that mapping is canceled. Once these connections have been struck from the connection map, all remaining validated connections are recorded as filled.

An additional tasking of this cancellation process concerns whether the mapping of either contour resulted in all coordinates defining that contour being included in the mapping. In that case, all other possible pairings with the completely mapped contour are canceled. This is accomplished by zeroing the overlap on that contour's row or column of the overlap table.

3.2.3.3. Connection Formation

When the above steps have been completed for a pair of contours, the remaining process of generating the appropriate line segments is relatively simple. The final coordinate mapping for the inner contour is examined for continuous segments of validated connections. When a continuous segment is defined, the beginning and ending coordinates of that segment (for both the inner and outer contours) are used as boundary pointers for connection formation. The coordinates in between those pointers are stepped through one at a time by a process whose purpose is to generate the minimum area triangular surface patch, as defined in our introduction. The surface patch is formed by using a line segment from one contour as the triangle's base, and a coordinate from the other contour for the triangle's third point. The minimum area selection is accomplished by a procedure that chooses the next line segment between the contours that is both the shortest and within the mapping specified for the two contours. This is identical to the heuristic used in [1]. Differing coordinate rates between the two contours are taken care of by using the coordinate ratio (from the continuity tolerance factor) between the contours. This ratio allows the process to generate several line segments emanating from a single coordinate where there is a coordinate rate differential between two mapped contours. The lines generated by this procedure for Figures 3.4

and 3.6 are shown in Figures 3.5 and 3.7, respectively.

3.2.4. Form the Coordinate Mapping: Exterior to Exterior

We begin the exterior to exterior mapping process at the same point of the algorithm where we departed in the description of the interior to interior mapping process. In keeping with our ordering criteria for mapping contour pairs, we examine the contour pair requiring an exterior to exterior mapping which has the highest overlap percentage in the overlap table. All remaining steps of the algorithm are carried out on this pair before the next pair of exterior to exterior contours, in largest to smallest overlap area, is considered.

In Figure 3.13, we are presented with an enlarged view of the bounding box overlap area of the contour pairing (1,1) - (2,2) of Figure 3.6. This area of overlap contains all of the coordinates from both contours which will be involved in the connection mapping. The first operation performed on an exterior to exterior mapped overlap pair is the determination of the set of coordinates in both contours that is within the overlap area. The contour with the smaller number of coordinates in the overlap area is used in the formation of a connection mapping between the contour with the larger number of coordinates in the overlap area. The basis for this connection map is the determination for each coordinate (in the smaller coordinate set contour) of the coordinate in the other contour coordinate set that is the shortest distance away. This determination is a simpler version of the distance minimizing process for connection set assignment of interior to interior mappings. The product of this process is the connection map for the pair of contours. The use of continuity sets is not necessary for exterior to exterior mappings due to the relatively small number of coordinates which comprise the connection set.

Once we have generated this connection set, we use the same mapping cancellation and connection formation procedures as described for the interior to interior mappings. The connection formation procedure again uses the connection set mapping to find continuous segments of validated coordinate assignments. The continuous segment thus defined is used to form triangular surface patches for all line segments and coordinates within that segment. The final connection formation for the exterior to exterior mappings, (1,1) - (2,2) and (2,1) - (2,2) of Figure 3.6, are

shown in Figure 3.7.

4. Algorithm Heuristics and Limitations

In the preceding section, we presented an explanation of our algorithm for surface construction. Particular attention was devoted to our algorithm's handling of the multiple contours per plane and partial contour mapping problems. It must be emphasized, however, that our algorithm does not provide a complete solution for all sets of contour surface data. In this section, we investigate some of the limitations of our algorithm. In order to do that, we must first discuss the heuristics employed by that algorithm.

4.1. Heuristics

Our algorithm utilizes three heuristics which are essential for the correct connection of planar contours. These heuristics were presented briefly in the last section, but we feel it is necessary to explain more fully their application and interaction regarding the contour mapping problem.

4.1.1. Overlap Percentage Minimum

In step two of our algorithm, we determine the percentage of overlap between contours on adjacent planes. These percentages are then considered in a consistency check for matching contour interior specifications. The heuristic in question, the overlap percentage minimum, is applied in the final phase of this contour pairing procedure. Contour pairs having an overlap percentage value above the overlap percentage minimum, with matching interior specifications, are designated for interior to interior mapping. Contour pairs having non-zero percentages below the overlap percentage minimum, with non-matching interior specifications, are designated for exterior to exterior mapping. All other contour pairs are disregarded.

The value we have utilized for the overlap percentage minimum is ten percent. We found, through experimentation, that the assignment of this value resulted in the greatest number of correct contour pairings. Some contour pairs which should be mapped, however, are disregarded

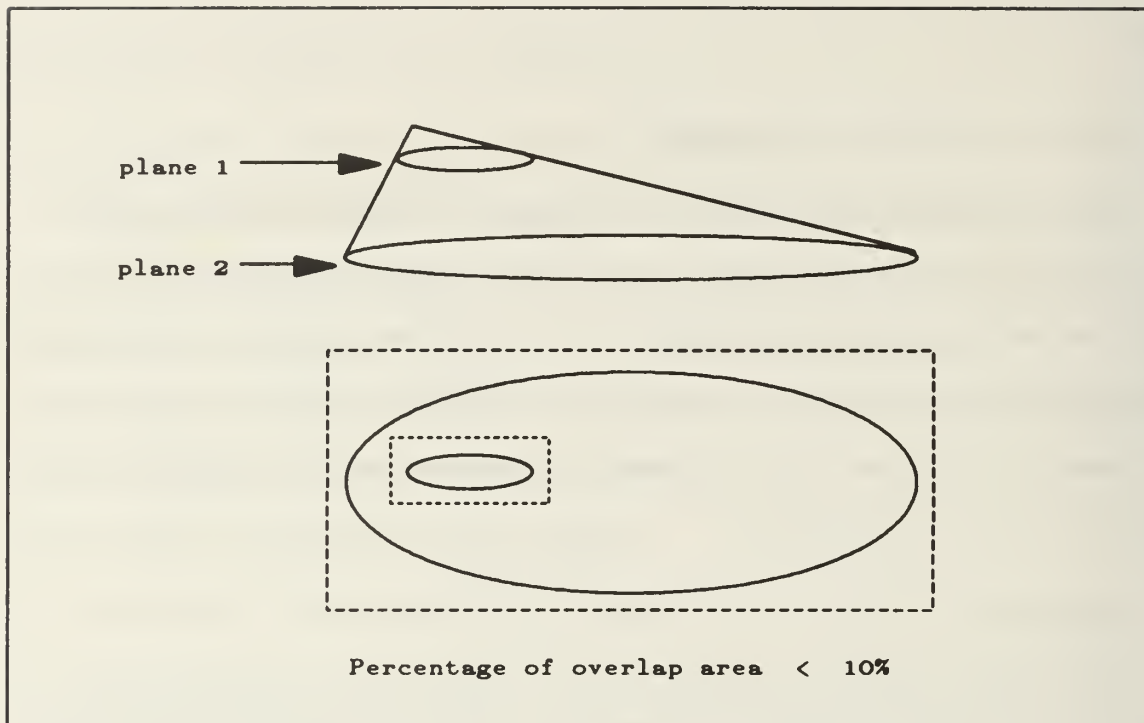


Fig. 4.1 - Example of a contour pair which should be mapped, but would be disregarded due to overlap percentage below the minimum.

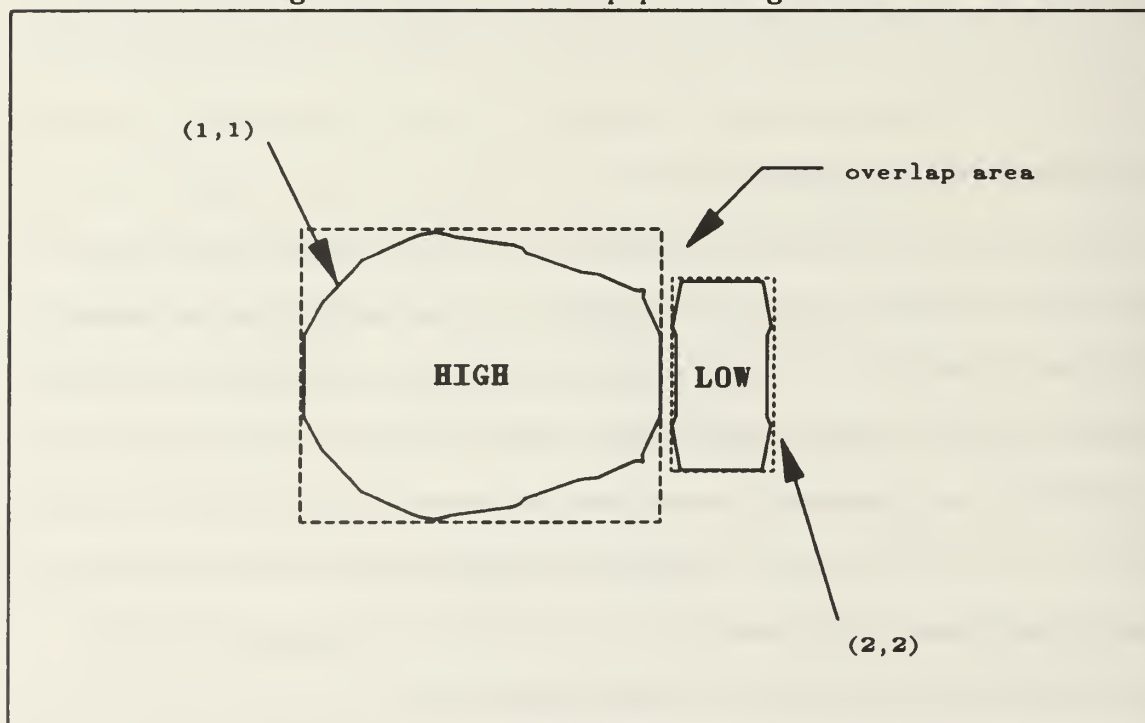


Fig. 4.2 - Example of contours' 2D bounding boxes created strictly from the min and max X and Y coordinates. Resulting overlap = 0.

for mapping because of this selection (of 10%) for the overlap percentage minimum. In Figure 4.1, we are presented with an example of such a situation. In that figure, we have a pair of contours with matching interior specifications (HIGH:HIGH), and having an overlap percentage less than ten percent. By our heuristic, this contour pair would not be considered for mapping, and would remain unconnected.

One possible solution to this problem would be a mechanism which used a relaxation procedure to force a mapping between the pair of contours. This mechanism could be selected by the user to designate contour pairs for mapping which would otherwise be disregarded. If applied to the mapping situation of Figure 4.1, an appropriate connection could be generated.

4.1.2. Boundary Tolerance Percentage

The next heuristic to be discussed comes into play in the initial two steps of our algorithm. Specifically, the two operations involved are the determination of contour item two-dimensional bounding box values, and the usage of those values for overlap determination. As previously discussed, exterior to exterior contour mappings are indicated for pairs of contours with a low percentage of overlap and non-matching interior specifications. In the initial development of our algorithm, we utilized the minimum and maximum X and Y coordinates of the contour to describe its bounding box. We found, however, that in the majority of cases, these values resulted in zero percentage of overlap between contours which should be mapped. An example of this limiting of bounding box values can be seen in Figure 4.2. In that figure, we are presented with the contour pair from Figure 3.13. In this example, it can be seen that limiting the bounding boxes for these two contours to their respective minimum and maximum X and Y coordinate values results in zero percentage of overlap. This is an unsatisfactory situation since the contours should be mapped.

To remedy this situation, we adjust the bounding box values by a percentage to promote mappings in situations similar to that of Figure 4.2. Once again, we are presented with the opportunity to utilize a relaxation procedure, prompted via user intervention, for mapping situations not included by this heuristic. A mechanism could be provided allowing the user to

designate the bounding boxes for individual contours, and thereby force a mapping between the desired set of contours.

4.1.3. Tolerance Multiplier

In an interior to interior mapping situation, a tolerance factor is used for the determination of the initial continuity set assignments. This tolerance factor is a ratio of the number of coordinates in the outer contour divided by the number of coordinates in the inner contour times a window value. The window value is a constant which we found necessary for the selection of appropriate mapping connections. We chose to utilize a tolerance factor in this step of our algorithm, as well as in the connection formation procedure, because it provides an inexpensive means for restricting the search space in the selection of mapping connections.

4.2. Limitations

In section 3, we demonstrated the capabilities of our algorithm, with emphasis on its handling of the problems of multiple contours per plane and partial contour mappings. We have found, however, that there exist contour mapping situations which cannot be handled by our algorithm.

The first mapping situation concerns simple branching of one contour on one plane to two or more contours on an adjacent plane (see Figure 2.3). In this situation, we found that the application of our algorithm produces an incomplete contour mapping due to missing data. One possible solution to this mapping problem is the inclusion of a procedure for creating an introduced node similar to that described in [1]. This special case procedure could be selected automatically, or initiated via user interaction.

The next limitation of our algorithm manifests itself in situations where highly convoluted contours, with extreme narrowings, are mapped interior to interior. The problem here is due to the interior to interior algorithm's dependence on the overlap region bounding box's center coordinate for the tentative coordinate mapping. For the portion of the contour near the center coordinate, the tentative coordinate is fairly good. For the portion of a contour on the other side of a

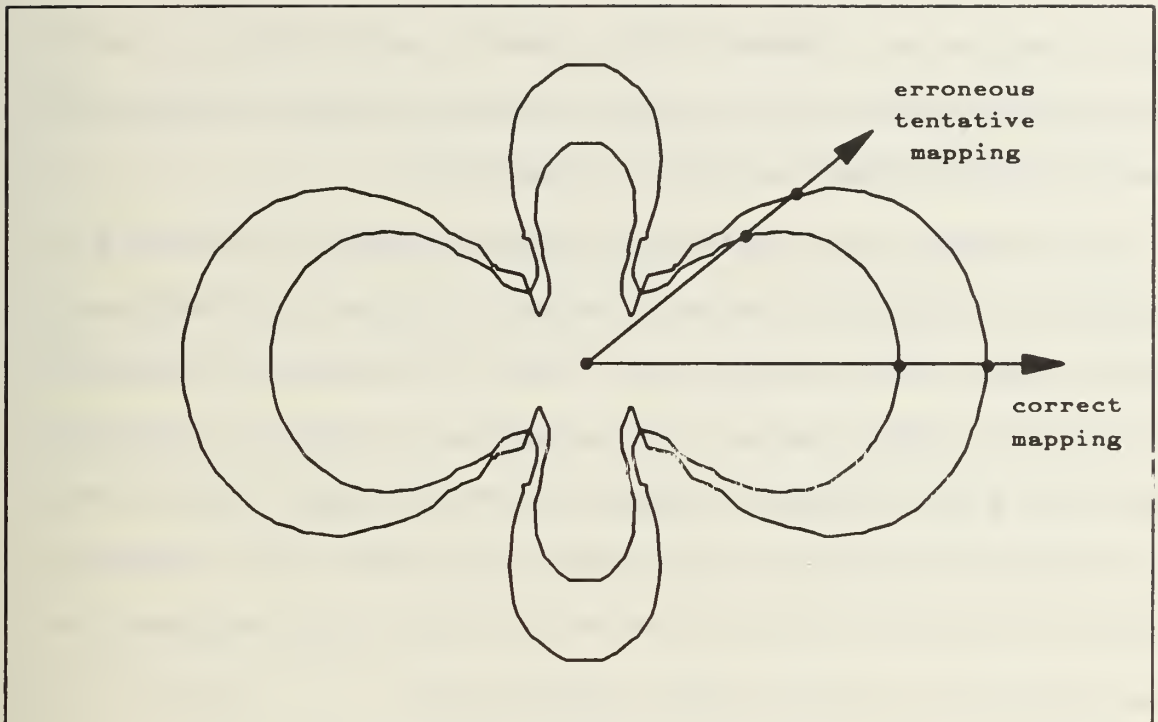


Fig. 4.3 - Example of situation resulting in an erroneous tentative coordinate mapping where contour segment becomes near parallel with the tentative connection vector.

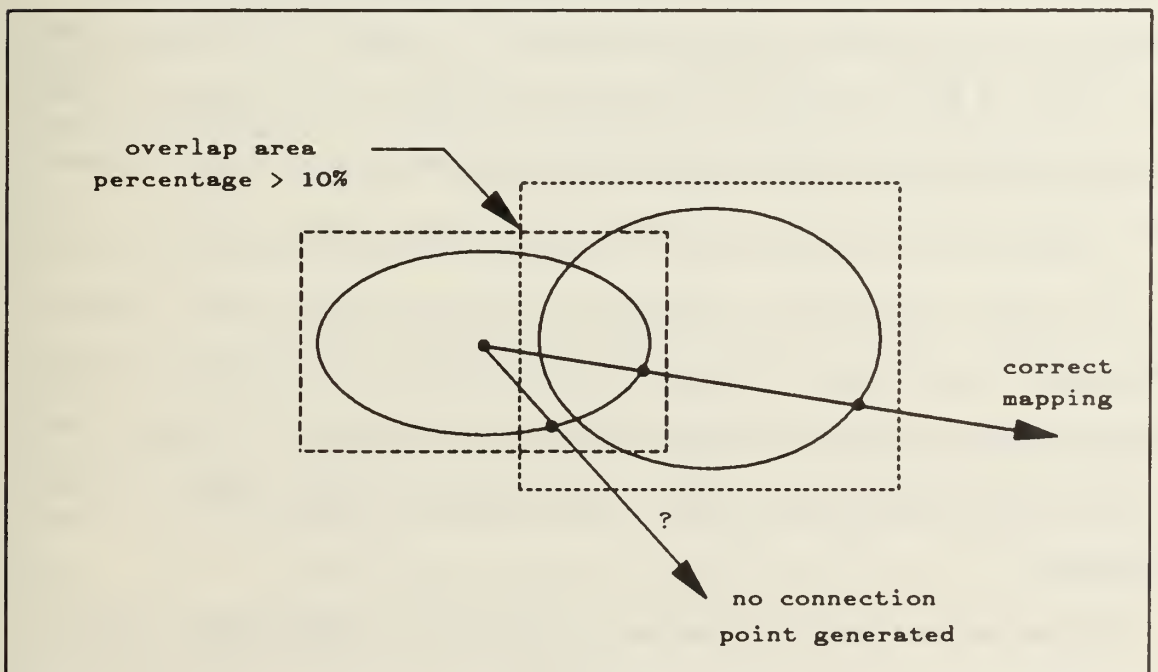


Fig. 4.4 - Example of a situation where two contours are mapped interior to interior which would result in an incomplete mapping.

narrowing, where the center coordinate is no longer central, the tentative mapping is erroneous. The problem comes when the tentative mapping is so bad that the continuity recognition procedure fails, and contour segments are incorrectly left unconnected.

The solution to this problem is fairly simple and within the purview of our algorithm. If the convoluted contour is segmented at the extreme narrowings, it is possible to treat each open segment of the original contour as a separate contour. Using the original algorithm, we can generate centers for each new contour, and hence coordinate mappings, which result in a more correct approximation of the original three-dimensional object. The only capability lacking from our present algorithm is a mechanism for partitioning the original convoluted contour. This mechanism could be either user specified or automatic. The user specified option is favored due to the computational expense involved for automatic contour segmentation.

Another limitation of our algorithm also concerns interior to interior mappings. In situations where sections of a contour tend to be near parallel with the vector drawn from the center coordinate of the inner contour, erroneous mappings result. An example of this situation can be seen in Figure 4.3. For those segments of the outer contour which are nearly perpendicular to the tentative connection vector, an appropriate connection map is generated. As the contour segment becomes more parallel to this vector, the tentative connections generated begin to falter.

The remedy to this problem is very similar to that for the previous situation involving highly convoluted contours with extreme narrowings. Segmentation of the original contour into several open segments, which can be mapped separately, would greatly improve the quality of the tentative coordinate mapping. Once again, user intervention is the preferred method of contour segmentation.

The final problem situation to be discussed concerns interior to interior mappings where the inner contour is not contained in the outer contour. This situation would result from contour data taken from a torus, such as a doughnut. An example is illustrated in Figure 4.4. The problem with this mapping situation results from the use of the tentative connection vector emanating from the center of the inner contour. Since the center coordinate of the inner contour is displaced

from the center coordinate of the outer contour, tentative mappings are generated only for that section of the outer contour which is on the same side of the tentative connection vector (see Figure 4.4). The net result is a partial mapping of two contours which should be totally connected.

A practical solution to this mapping problem, which could be readily adapted to our algorithm, is described in [1]. In mapping situations where contours to be mapped are not mutually centered, Christiansen recommends a translation procedure onto a unit square, centered at (0,0). The principle of this process is to translate the two contours in such a manner that they become mutually centered within the unit square. Application of the interior to interior algorithm at this point would result in the desired mappings. Tentative mappings would be generated for the contours' original coordinates, thus allowing the appropriate connections to be formed in the final step of the algorithm.

5. Conclusions

It has been the goal of this paper to describe a new algorithm for the surface construction of a three-dimensional object from a set of that object's planar contours. The greatest part of this paper has been devoted to the capabilities of our algorithm, specifically, its handling of the multiple contours per plane and partial contour mapping problems. We have included a discussion of the limitations encountered thus far by our algorithm for specific problem mapping situations.

In view of the limitations presented, we must comment that our algorithm does not, in its present form, provide a complete solution to the contour mapping problem. Further development is required to alleviate the problem areas discussed in section 4. It is probable, however, that the correction of these algorithmic shortcomings will not ensure a complete solution to the contour mapping problem. We foresee that in some situations either user intervention or an alternative approach may be required.

6. References

1. Christiansen, H. N. and Sederberg, T. W. "Conversion of Complex Contour Line Definitions into Polygonal Mosaics," *Computer Graphics: A Quarterly Report of SIGGRAPH-ACM*, Vol. 12 (August 1978), pp. 187-192.
2. Fuchs, H., Kedem, Z. M., and Uselton, S. P. "Optimal Surface Reconstruction from Planar Contours," *Communications of the ACM*, Vol. 20 (October 1977), pp. 693-702.
3. Ganapathy, S. and Dennehy, T.G. "A New General Triangulation Method for Planar Contours," *Computer Graphics: A Quarterly Report of SIGGRAPH-ACM*, Vol. 16 (July 1982), pp. 69-74.
4. Shantz, M. "Surface Definition for Branching, Contour-Defined Objects," *Computer Graphics: A Quarterly Report of SIGGRAPH-ACM*, Vol. 15 (July 1981), pp. 242-266.

7. Appendix - Pseudocode of the Surface Construction Algorithm

FACEIT

```
{
    Input the coordinates for two adjacent planes. Make a local copy of the coordinates.

    DELINEATE_INVENTORY
    {
        Take inventory of the contours in the coordinate sets. This inventory determines
        the total number of contours for each plane and records where each contour begins
        and ends.
    }

    TYPE_INVENTORY
    {
        We determine the contour type of each contour on each plane. There are three possible types: CLOSED_LOOP, OPEN_SEGMENT, and SINGLE_POINT.
    }

    BOUND_INVENTORY
    {
        Determine the rectangular, two-dimensional boundary of each contour. Increase those boundaries by a constant to increase the possibility of detecting appropriate exterior to exterior mappings.
    }

    INTERIOR_DETERMINATION
    {
        Determine whether the interior of each contour is HIGH or LOW valued with respect to the current contour level. This value can be assigned interactively in cases where the information to make this determination is not available. These values are used in a consistency check for selection of contour pairs for mapping.
    }

    OVERLAP_DETERMINATION
    {
        Compute the overlap table for the contours of both planes. The values in the table are the percentage of overlap for each possible contour pair on the adjacent planes. If there is no overlap, value of 0.0 is recorded.

        Contour mapping types are also assigned at this step of the algorithm. Contour pairs with a HIGH percentage of overlap, matching interior specifications (HIGH:HIGH, LOW:LOW, or INDETERMINATE:anything) are assigned interior to interior type mapping. Those pairs with a non-zero overlap percentage, below 10%, with non-matching interiors are assigned exterior to exterior mappings. All other contour pairings are zeroed.
    }
}
```

CONNECTION_DETERMINATION

```
{  
    This step of the algorithm orders the pairs to be mapped, and forms connections for  
    the assigned types of contour mappings. This step is detailed below.  
}  
  
} /* end of FACEIT */
```

CONNECTION_DETERMINATION

```
{  
    while (TRUE)  
    {  
        Find the largest overlap percentage in the overlap table. If the largest value = 0.0  
        the QUIT.  
  
        If the contour mapping indicated by this largest overlap value is exterior to exterior  
        {  
            EXTERIOR_TO_EXTERIOR_MAPPING  
            {  
                Determine the set of coordinates in each contour that are in the over-  
                lap area.  
  
                For the contour of the overlap pair that has the least number of coor-  
                dinates, find the minimum distanced coordinate of the other contour.  
  
                Assign all coordinates within the overlap region to the connection set.  
            }  
        } * endif was exterior to exterior mapping */  
        else  
        {  
            /* Perform an interior to interior mapping */  
  
            INTERIOR_TO_INTERIOR_MAPPING  
            {  
                Determine which contour of the pair is interior. This assignment is  
                based upon which contours' bounding box is smallest.  
  
                Compute the center coordinate of the inner contour's bounding box.  
                Check to make sure that this point is inside the contour. If it is not,  
                the contour needs to be partitioned.  
  
                For each coordinate of the inner contour, determine the coordinate of  
                the outer contour which is closest to a vector drawn from the center  
                coordinate through the coordinate of the inner contour. Store the  
                coordinate as the connection map coordinate for the inner contour.  
                Also, record the mapped distance from each inner coordinate to its  
                mapped outer coordinate.
```

RECOGNIZE_CONTINUITY

```
{  
    /* Determine continuity sets in the two contours using the con-  
    nection map and associated distances. */
```

INITIAL_CONTINUITY_SETS

```
{  
    Assign the coordinates of the connection map to a con-  
    tinuity set based upon whether each consecutive coordi-  
    nate is within a coordinate tolerance factor. This toler-  
    ance factor is a ratio of the number of coordinates in the  
    outer contour divided by the number of coordinates in  
    the inner contour.  
}
```

INITIAL_SET_DISTANCE_RANGES

```
{  
    Determine the minimum and maximum distance ranges  
    for each of the continuity sets.  
}
```

CONTINUITY_SET_MERGE

```
{  
    Merge any continuity sets that have overlapping distance  
    ranges, maintaining the distance range for any merged  
    set.  
}
```

CONNECTION_SET_ASSIGNMENT

```
{  
    Assign coordinate connections for the coordinates of the  
    merged continuity set that contains the smallest dis-  
    tance. All other continuity sets are left unconnected.  
}
```

```
} /* end of RECOGNIZE_CONTINUITY */
```

```
} /* end of INTERIOR_TO_INTERIOR_MAPPING */
```

```
} /* endelse was an interior to interior mapping */
```

MAPPING_CANCELLATION

```
{  
    Examine the coordinate mappings for which a connection has been assigned.  
    If either of the two coordinates, inner contour or outer contour, has been used  
    in a previous, higher priority mapping for this pair of planes, that coordinate  
    mapping is canceled. Once these filled connections have been struck from the  
    connection map, all remaining validated connections are recorded as filled.  
}
```


CONNECTION_FORMATION

{

Generate the connections for the validated coordinate map. This is accomplished by stepping through the connection map and forming coordinate connections where indicated. In between coordinates, those not directly mapped but within the tolerance factor for the connection mapping, are also added to the picture. The goal of the connection process is to form minimum area triangular surface patches.

}

} /* end while (TRUE) */

} /* end of CONNECTION_DETERMINATION */

Distribution List for Papers Written by Michael J. Zyda

Defense Technical Information Center,
Cameron Station,
Alexandria, VA 22314 2 copies

Library, Code 0142
Naval Postgraduate School,
Monterey, CA 93943 2 copies

Center for Naval Analyses,
2000 N. Beauregard Street,
Alexandria, VA 22311

Director of Research Administration,
Code 012,
Naval Postgraduate School,
Monterey, CA 93943

Dr. Henry Fuchs,
208 New West Hall (035A),
University of North Carolina,
Chapel Hill, NC 27514

Dr. Kent R. Wilson,
University of California, San Diego
B-014,
Dept. of Chemistry,
La Jolla, CA 92093

Dr. Guy L. Tribble, III
900 Waverly St.
Palo Alto, California 94301

Bill Atkinson.
Apple Computer,
20525 Mariani Ave,
Cupertino, CA 95014

Dr. Victor Lesser,
University of Massachusetts, Amherst
Dept. of Computer and Information Science,
Amherst, MA 01003

Dr. Gunther Schrack.
Dept. of Electrical Engineering,
University of British Columbia,
Vancouver, B.C.. Canada V6T 1W5

Dr. R. Daniel Bergeron,
Dept. of Computer Science,
University of New Hampshire,
Durham, NH 03824

Dr. Ed Wegman,
Division Head,
Mathematical Sciences Division,
Office of Naval Research,
800 N. Quincy Street,
Arlington, VA 22217-5000

Dr. Gregory B. Smith,
ATT Information Systems,
190 River Road,
Summit, NJ 07901

Dr. Lynn Conway,
University of Michigan,
263 Chrysler Center,
Ann Arbor, MI 48109

Dr. John Lowrance,
SRI International,
333 Ravenswood Ave,
Menlo Park, CA 94025

Dr. David Mizell,
Office of Naval Research,
1030 E. Green St.
Pasadena, CA 91106

Dr. Richard Lau,
Office of Naval Research,
Code 411,
800 N. Quincy St.
Arlington, VA 22217-5000

Dr. Y.S. Wu,
Naval Research Laboratory,
Code 7007,
Washington, D.C. 20375

Dr. Joel Trimble,
Office of Naval Research,
Code 251,
Arlington, VA 22217-5000

Robert A. Ellis,
Calma Company,
R & D Engineering,
525 Sycamore Dr., M/S C51O
Milpitas, CA 95035-7489

Dr. James H. Clark,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Edward R. McCracken,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Shinji Tomita,
Dept. of Information Science,
Kyoto University,
Sakyo-ku, Kyoto, 606, Japan

Hiroshi Hagiwara,
Dept. of Information Science,
Kyoto University,
Sakyo-ku, Kyoto, 606, Japan

Dr. Alain Fournier,
Dept. of Computer Science,
University of Toronto,
Toronto, Ontario, Canada
M5S 1A4

Dr. Andries Van Dam,
Dept. of Computer Science,
Brown University,
Providence, RI 02912

Dr. Brian A. Barsky,
Berkeley Computer Graphics Laboratory,
Computer Sciences Division,
Dept. of Electrical Engineering and Computer Sciences,
University of California,
Berkeley, CA 94720

Dr. Ivan E. Sutherland,
Carnegie Mellon University,
Pittsburg, PA 15213

Dr. Turner Whitted,
New West Hall (035A),
University of North Carolina,
Chapel Hill, NC 27514

Dr. Robert B. Grafton,
Office of Naval Research,
Code 433,
Arlington, Virginia 22217-5000

Professor Eihachiro Nakamae,
Electric Machinery Laboratory,
Hiroshima University,
Higashihiroshima 724, Japan

Carl Machover,
Machover Associates,
199 Main Street,
White Plains, New York 10601

Dr. Buddy Dean,
Naval Postgraduate School,
Code 52, Dept. of Computer Science,
Monterey, California 93943

Earl Billingsley,
43 Fort Hill Terrace,
Northhampton, MA 01060

Dr. Jan Cuny,
University of Massachusetts, Amherst
Dept. of Computer and Information Science,
Amherst, MA 01003

Robert Lum,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Jeff Hausch,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Lt. Robert A. Walker,
Naval Sea Systems Command (SEA 61YM),
Department of the Navy,
Washington, DC 20362-5101

Dr. Barry L. Kalman,
Washington University,
Department of Computer Science,
Box 1045,
St. Louis, Missouri 63130

Dr. Wm. Randolph Franklin,
Electrical, Computer, and Systems Engineering Department,
Rensselaer Polytechnic Institute,
Troy, New York 12180-3590

Dr. Gershon Kedem,
Microelectronics Center of North Carolina,
PO Box 12889,
3021 Cornwallis Road,
Research Triangle Park,
North Carolina 27709

Dr. Branko J. Gerovac,
Digital Equipment Corporation,
150 Locke Drive LMO4/H4, Box 1015
Marlboro, Massachusetts 01752-9115

Robert A. Schumacker,
Evans and Sutherland,
PO Box 8700,
580 Arapeen Drive,
Salt Lake City, Utah 84108

R. A. Dammkoehler,
Washington University,
Department of Computer Science,
Box 1045,
St. Louis, Missouri 63130

Dr. Lynn Ten Eyck.
Interface Software.
79521 Highway 99N,
Cottage Grove, Oregon 97424

Toshiaki Yoshinaga,
Hitachi Works, Hitachi Ltd.
1-1, Saiwaicho 3 Chome,
Hitachi-shi, Ibaraki-ken,
317 Japan

Takatoshi Kodaira,
Omika Works, Hitachi Ltd.
2-1, Omika-cho 5-chome,
Hitachi-shi, Ibaraki-ken,
319-12 Japan

Atsushi Suzuki.
Hitachi Engineering, Co. Ltd.
2-1, Saiwai-cho 3-Chome,
Hitachi-shi, Ibaraki-ken,
317 Japan

Toshiro Nishimura,
Hitachi Engineering, Co. Ltd.
2-1, Saiwai-cho 3-Chome,
Hitachi-shi, Ibaraki-ken,
317 Japan

Dr. John Staudhammer.
Dept. of Electrical Engineering.
University of Florida.
Gainesville, Florida 32611

Dr. Lewis E. Hitchner,
Computer and Information Science Dept.
237 Applied Science Building,
University of California at Santa Cruz,
Santa Cruz, California 95064

Dr. Pat Mantey,
Computer Engineering Department,
University of California at Santa Cruz,
Santa Cruz, California 95064

Dr. Walter A. Burkhardt,
University of California, San Diego
Dept. of Computer Science,
La Jolla, California 92093

P. K. Rustagi,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Peter Broadwell,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Norm Miller,
Silicon Graphics, Inc.
2011 Stierlin Road,
Mountain View, CA 94043

Dr. Tosiyasu L. Kunii,
Department of Information Science,
Faculty of Science,
The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113,
Japan

Dr. Kazuhiro Fuchi,
Institute for New Generation Computer Technology.
Mita-Kokusai Building 21FL.
1-4-28 Mita, Minato-ku, Tokyo 108, Japan

Tony Loeb,
Silicon Graphics, Inc.
1901 Avenue of the Stars,
Suite 1774,
Los Angeles, CA 90067

Kevin Hammons,
NASA AMES-Dryden Flight Research Facility.
PO Box 273,
Mail Stop OFI,
Edwards, California 93523

Sherman Gee,
Code 221,
Office of Naval Technology,
800 N. Quincy St.
Arlington, VA 22217

Dr. J.A. Adams,
Department of Mechanical Engineering,
US Naval Academy,
Annapolis, MD 21402

Dr. David F. Rogers,
Dept. of Aerospace Engineering,
US Naval Academy,
Annapolis, MD 21402

Dr. Robert F. Franklin,
Environmental Research Institute of Michigan,
PO Box 8618,
Ann Arbor, MI 48107

LT Mark W. Hartong,
900 Cambridge Dr 17,
Benicia, CA 94510

Capt. Mike Gaddis,
DCA/JDSSC/C720,
1860 Wiehle Ave
Reston, VA 22090

Lt. Cdr. Patrick G. Hogan, USN
102 Borden Avenue,
Wilmington, North Carolina 28403

Dr. Edwin Catmull,
LucasFilm,
PO Box 2009,
San Rafael, CA 94912

Dr. John Beatty,
Computer Science Department,
University of Waterloo,
Waterloo, Ontario,
Canada N2L 3G1

Dr. James Foley,
George Washington University,
Dept. of Electrical Engineering and Computer Science,
Washington, D.C. 20052

Dr. Donald Greenberg,
Cornell University,
Program of Computer Graphics,
Ithaca, NY 14853

Dr. Leo J. Guibas,
Systems Research Center,
Digital Equipment Corporation,
130 Lytton Avenue,
Palo Alto, CA 94301

Dr. S. Ganapathy,
Ultrasonic Imaging Laboratory,
Dept. of Electrical and Computer Engineering,
University of Michigan,
Ann Arbor, MI 48109

Dr. Hank Christiansen,
Brigham Young University,
Dept. of Civil Engineering,
368 Clyde Bldg.
Provo, Utah 84602

Dr. Thomas A. DeFanti,
Dept. of Electrical Engineering & Computer Science,
University of Illinois at Chicago,
Box 4348.
Chicago, IL 60680

Dr. Lansing Hatfield,
Lawrence Livermore National Laboratory,
7000 East Avenue,
PO Box 5504, L-156,
Livermore, CA 94550

DUDLEY KNOX LIBRARY



3 2768 00347419 8